



TUGAS AKHIR - TE 145561

***SKATEBOARD ELEKTRIK DENGAN SMARTPHONE SEBAGAI
PENGATUR KECEPATAN***

Stefanus Aldi Atmaja
NRP 2213030060
Annisa Yunitasari F
NRP 2213030076

Dosen Pembimbing
Rudy Dikairono, ST., MT.

PROGRAM STUDI D3 TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT - TE 145561

ELECTRIC SKATEBOARD USING SMARTPHONE AS SPEED CONTROLLER

Stefanus Aldi Atmaja

NRP 2213030060

Annisa Yunitasari F

NRP 2213030076

Advisor

Rudy Dikairono, ST., MT.

ELECTRICAL ENGINEERING D3 STUDY PROGRAM

Faculty of Industrial Technology

Institut Teknologi Sepuluh Nopember

Surabaya 2016

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "*Skateboard Elektrik Dengan Smartphone Sebagai Pengatur Kecepatan*" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Mahasiswa 1



Stefanus Aldi Atmaja
NRP. 2213 030 060

Surabaya, 30 Mei 2016

Mahasiswa 2



Annisa Yunitasari F
NRP. 2213 030 076

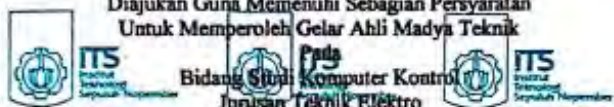
-----Halaman ini sengaja dikosongkan-----



SKATEBOARD ELEKTRIK DENGAN SMARTPHONE SEBAGAI PENGATUR KECEPATAN



Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya Teknik



Padu
Bidang Studi Komputer Kontrol
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember



SURABAYA
30 MEI, 2016



-----Halaman ini sengaja dikosongkan-----

SKATEBOARD ELEKTRIK DENGAN SMARTPHONE SEBAGAI PENGATUR KECEPATAN

Nama 1 : Stefanus Aldi Atmaja
Nama 2 : Annisa Yunitasari F
Pembimbing : Rudy Dikairono, ST., MT.

ABSTRAK

Seiring dengan perkembangan zaman dan teknologi, inovasi semakin luas. *Skateboard* tidak hanya digunakan sebagai sarana bermain ataupun olahraga. *Skateboard* mulai dialihkan fungsinya sebagai alat transportasi yang ramah lingkungan. Oleh karena itu, timbul ide merancang *skateboard* elektrik untuk mengatasi kemacetan jalan raya serta transportasi pribadi yang rendah polusi.

Penelitian ini membahas tentang kontrol kecepatan motor pada *skateboard*. *Skateboard* elektrik dikendalikan melalui sensor *accelerometer* pada *smartphone*. *Smartphone* akan difasilitasi dengan aplikasi yang *user friendly* untuk mengoperasikan kecepatan motor dc. Data yang terbaca oleh aplikasi *smartphone* akan dikirimkan pada mikrokontroler dengan melalui *bluetooth*. Data *accelerometer* yang terbaca dalam bentuk presentase dimana nilai maksimal 100% sama dengan 90° perpindahan. Sensor *accelerometer* yang terbaca hanya memanfaatkan perubahan nilai X axis. Jika nilai *accelerometer* bernilai minus maka *skateboard* tidak akan bergerak. Mikrokontroler akan mengolah data yang diterima untuk mengontrol kecepatan motor dc. Sehingga pengguna dapat mengatur kecepatan motor sesuai dengan yang diinginkan.

Skateboard ini dapat melaju dengan kecepatan maksimal 10 km/jam dengan beban maksimal 60 kg. Pada alat ini koneksi *bluetooth* dapat dijangkau dengan jarak maksimal 37 km. Data *accelerometer* yang diperoleh masih terdapat error, error maksimal pada busur 18° yaitu dengan nilai error sebesar 45%. Dengan adanya *skateboard* elektrik ini, pengguna tidak perlu mengoperasikan *skateboard* secara manual karena dengan *smartphone* dapat membantu pergerakan *skateboard*. Tidak hanya mempermudah pengguna, namun *skateboard* ini juga ramah lingkungan.

Kata Kunci : *Skateboard* elektrik, *smartphone*, kontrol kecepatan

-----Halaman ini sengaja dikosongkan-----

ELECTRIC SKATEBOARD USING SMARTPHONE AS SPEED CONTROLLER

Name 1 : Stefanus Aldi Atmaja
Name 2 : Annisa Yunitasari F
Advisor : Rudy Dikairono, ST., MT.

ABSTRACT

Along with the times and technology development, innovation becomes more wide. Skateboard not only used for fun sport, but now also used as a transportation which causes less pollution. Therefore, an idea is created to design an electric skateboard to decrease the traffic jam risk and lower the pollution caused by vehicle transportation.

The research is focusing on controlling the speed of skateboard. Skateboard is controlled by smartphone using accelerometer placed on it. There will be an user-friendly application on smartphone to adjusting the speed of the skateboard. The speed data on smartphone will be transmitted, and received by the microcontroller on skateboard via bluetooth. Accelerometer sensor is read form the percentage which the maximum value of 100% is equal to 90° displacement. The accelerometer sensor reading only changes in the value of X axis. If the accelerometer value minus the skateboard will not move. Microcontroller will process the data so that the speed of DC motor of skateboard could change related to the data received.

Skateboard can move with maximum speed of 10 km/hour, with the maximum weight is 60 kg. Bluetooth can be used in maximum distance about 37 km. Accelerometer sensor obtained has an error, maximum error at position 18° is the error value by 45%. With this electric skateboard, the user doesn't have to step and move the skateboard manually, since the smartphone could handle the speed control system. Not only facilitate the user, but this skateboard also eco-friendly.

Keywords : Electric skateboard, Smartphone, speed control

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma – 3 pada Bidang Studi Computer Control, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

SKATEBOARD ELEKTRIK DENGAN SMARTPHONE SEBAGAI PENGATUR KECEPATAN

Dalam Tugas Akhir ini dirancang *electric skateboard* yang merupakan alat transportasi jarak dekat. Pengguna *skateboard* dapat mengendalikan kecepatan motor dengan melalui *smartphone*.

Penulis mengucapkan terima kasih kepada keluarga yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Rudy Dikairono, ST., MT. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini, Bapak Andri Ashafani, ST., M.Eng. selaku koordinator Tugas akhir yang selalu memberi semangat, dukungan serta doa agar tugas akhir dapat terselesaikan dengan baik. Teman – teman angkatan 2013 selaku teman seperjuangan yang selalu memberikan doa, semangat, dan dukungannya kepada penulis. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 30 Mei 2016

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	Error! Bookmark not defined.
HALAMAN PENGESAHAN	Error! Bookmark not defined.
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xviii
DAFTAR TABEL	xxi
BAB I PENDAHULUAN	Error! Bookmark not defined.
1.1 Latar Belakang	Error! Bookmark not defined.
1.2 Permasalahan	Error! Bookmark not defined.
1.3 Batasan Masalah.....	Error! Bookmark not defined.
1.4 Tujuan	Error! Bookmark not defined.
1.5 Metodologi Penelitian	Error! Bookmark not defined.
1.6 Sistematika Laporan.....	Error! Bookmark not defined.
1.7 Relevansi	Error! Bookmark not defined.
BAB II TEORI DASAR	Error! Bookmark not defined.
2.1 Tinjauan Pustaka	Error! Bookmark not defined.
2.2 Mikrokontroler ATmega16	Error! Bookmark not defined.
2.3 Motor DC	Error! Bookmark not defined.
2.4 Sensor <i>Accelerometer</i>	Error! Bookmark not defined.
2.5 HC – 05	Error! Bookmark not defined.
2.6 Codevision AVR	Error! Bookmark not defined.
2.7 Baterai (<i>Accu</i>)	Error! Bookmark not defined.
2.8 IC LM 2596.....	Error! Bookmark not defined.
2.9 <i>Driver Motor</i>	Error! Bookmark not defined.
2.10 PWM (<i>Pulse Width Modulation</i>).	Error! Bookmark not defined.
BAB III PERANCANGAN DAN PEMBUATAN ALAT	Error! Bookmark not defined.

3.1 Perancangan <i>Hardware</i>	Error! Bookmark not defined.
3.1.1 Perancangan Mikrokontroler ATmega 16.....	Error! Bookmark not defined.
3.1.2 Rangkaian <i>Driver</i> Motor DC.....	Error! Bookmark not defined.
3.1.3 Rangkaian HC-05	Error! Bookmark not defined.
3.1.4 Rangkaian Catu Daya.....	Error! Bookmark not defined.
3.1.5 <i>Regulator</i>	Error! Bookmark not defined.
3.2 Perhitungan Torsi Roda	Error! Bookmark not defined.
3.3 Perancangan Motor DC	Error! Bookmark not defined.
3.4 Perancangan <i>Software</i>	Error! Bookmark not defined.
3.4.1 Perancangan Program pada <i>Software</i> CV AVRE.....	Error! Bookmark not defined.
3.4.2 Perancangan Program Kontrol Kecepatan Motor DC.....	Error! Bookmark not defined.
3.4.3 Perancangan Program Aplikasi <i>Smartphone</i>	Error! Bookmark not defined.
3.5 Perancangan <i>Hardware Skateboard</i> Elektrik.....	Error! Bookmark not defined.
BAB IV UJI COBA DAN ANALISA DATA Error! Bookmark not defined.	
4.1 Hasil Perancangan Alat.....	Error! Bookmark not defined.
4.2 Pengujian Motor DC	Error! Bookmark not defined.
4.3 Pengujian <i>Regulator</i>	Error! Bookmark not defined.
4.4 Pengujian <i>Output</i> Catu Daya	Error! Bookmark not defined.
4.5 Pengujian Sensor <i>Accelerometer</i>	Error! Bookmark not defined.
4.6 Pengujian RPM.....	Error! Bookmark not defined.
4.7 Pengujian <i>Output</i> Tegangan Mikrokontroler.....	Error! Bookmark not defined.
4.8 Pengujian <i>Output</i> Tegangan Motor DC.....	Error! Bookmark not defined.
4.9 Pengujian Arus pada Rangkaian Elektronik.....	Error! Bookmark not defined.
4.10 Pengambilan Data PWM.....	Error! Bookmark not defined.
4.11 Pengujian Jarak Komunikasi <i>Bluetooth</i>	Error! Bookmark not defined.
4.12 Pengujian Aplikasi dengan Menggunakan <i>Smartphone</i>	Error! Bookmark not defined.
BAB V PENUTUP	1
5.1 Kesimpulan	1
5.2 Saran	1
DAFTAR PUSTAKA	3
LAMPIRAN A	A-1
A.1. Tampilan Alat	A-1
LAMPIRAN B	B-1
B.1. <i>Listing</i> Program dengan CV AVR	B-1
LAMPIRAN C	C-1

C.1. <i>Datasheet Driver</i> Motor DC BTS7960	C-1
DAFTAR RIWAYAT HIDUP	D-1

-----Halaman ini sengaja dikosongkan-----

DAFTAR GAMBAR

HALAMAN

Gambar 2.1 Konfigurasi Pin ATmega 16	Error! Bookmark not defined.
Gambar 2.2 Bentuk Fisik ATmega16	Error! Bookmark not defined.
Gambar 2.3 Motor DC MY1025	Error! Bookmark not defined.
Gambar 2.4 Sumbu pada Sensor <i>Accelerometer</i>	Error! Bookmark not defined.
Gambar 2.5 Modul <i>Bluetooth</i> HC -05	Error! Bookmark not defined.
Gambar 2.6 CodeVision AVR	Error! Bookmark not defined.
Gambar 2.7 Pin Konfigurasi IC LM2596	Error! Bookmark not defined.
Gambar 2.8 <i>Driver</i> Motor BTS7960	Error! Bookmark not defined.
Gambar 2.9 Pin Konfigurasi <i>Driver</i> Motor BTS7960	Error! Bookmark not defined.
Gambar 2.10 <i>Pulse Width Modulation</i> .	Error! Bookmark not defined.
Gambar 3.1 Perancangan Sistem Secara keseluruhan	Error! Bookmark not defined.
Gambar 3.2 Skema Rangkaian Secara Keseluruhan	Error! Bookmark not defined.
Gambar 3.3 Rangkaian Mikrokontroler ATmega16	Error! Bookmark not defined.
Gambar 3.4 Skema Rangkaian <i>Driver</i> Motor BTS7960 dengan Mikrokontroler	Error! Bookmark not defined.
Gambar 3.5 Skema rangkaian HC-05 dengan mikrokontroler	Error! Bookmark not defined.
Gambar 3.6 Baterai 12 volt	Error! Bookmark not defined.
Gambar 3.7 Skema Rangkaian <i>Regulator</i> LM2596 dengan Mikrokontroler	Error! Bookmark not defined.
Gambar 3.8 Pemodelan Beban di Bidang Miring	Error! Bookmark not defined.
Gambar 3.9 Flowchart Sistem Secara Keseluruhan	Error! Bookmark not defined.
Gambar 3.10 Tampilan Awal CVAVR ..	Error! Bookmark not defined.
Gambar 3.11 Tampilan Membuka <i>Project</i> Baru CVAVR	Error! Bookmark not defined.
Gambar 3.12 Tampilan Dialog Box Code Wizard CVAVR	Error! Bookmark not defined.
Gambar 3.13 Tampilan Type <i>Chip</i> Mikrokontroler	Error! Bookmark not defined.
Gambar 3.14 Pengaturan Mikrokontroler dan <i>Clock</i>	Error! Bookmark not defined.
Gambar 3.15 Tampilan Konfigurasi <i>Port D</i> ATmega 16	Error! Bookmark not defined.
Gambar 3.16 Tampilan Pengaturan Komunikasi Serial	Error! Bookmark not defined.
Gambar 3.17 Tampilan Pengaturan Pada <i>Library Timers</i>	Error! Bookmark not defined.
Gambar 3.18 Tampilan Codewizard AVR	Error! Bookmark not defined.
Gambar 3.19 Tampilan Menyimpan File C Compiler	Error! Bookmark not defined.
Gambar 3.20 Tampilan Program Awal ..	Error! Bookmark not defined.
Gambar 3.21 Tampilan Website MIT App Inventor	Error! Bookmark not defined.

Gambar 3.22 Tampilan Awal MIT App Inventor**Error! Bookmark not defined.**

Gambar 3.23 Tampilan *Project Name***Error! Bookmark not defined.**

Gambar 3.24 Tampilan *Setting***Error! Bookmark not defined.**

Gambar 3.25 Tampilan Sub Menu Sensor**Error! Bookmark not defined.**

Gambar 3.26 Tampilan Block Editor.....**Error! Bookmark not defined.**

Gambar 3.27 Tampilan *Coding* Program**Error! Bookmark not defined.**

Gambar 3.28 Tampilan Menu *Build***Error! Bookmark not defined.**

Gambar 3.29 Tampilan *Barcode Link*.....**Error! Bookmark not defined.**

Gambar 3.30 Desain *Skateboard* Elektrik**Error! Bookmark not defined.**

Gambar 3.31 Perancangan Gear pada Roda**Error! Bookmark not defined.**

Gambar 4.1 Tampilan Pada Bagian Sampling *Skateboard***Error! Bookmark not defined.**

Gambar 4.2 Tampilan Pada Bagian Atas *Skateboard***Error! Bookmark not defined.**

Gambar 4.3 Tampilan *Skateboard* Secara Keseluruhan**Error! Bookmark not defined.**

Gambar 4.4 Tampilan Pada Bagian Bawah *Skateboard***Error! Bookmark not defined.**

Gambar 4.5 Grafik Pengujian Sensor *Accelerometer***Error! Bookmark not defined.**

Gambar 4.6 Pengujian Data Sensor *Accelerometer***Error! Bookmark not defined.**

Gambar 4.7 Grafik Perbandingan Nilai RPM dengan Beban $\pm 60\text{kg}$ dan Tanpa Beban**Error! Bookmark not defined.**

Gambar 4.8 Grafik Nilai Kecepatan dengan Beban $\pm 60\text{kg}$ **Error! Bookmark not defined.**

Gambar 4.9 Grafik Pengujian Output Tegangan Motor DC**Error! Bookmark not defined.**

Gambar 4.10. Pengambilan Data Arus Pada Rangkaian Skateboard Elektrik.**Error! Bookmark not defined.**

Gambar 4.11 Pengambilan Data Arus yang Menggunakan Rangkaian Pembagi Arus.**Error! Bookmark not defined.**

Gambar 4.12 Gambar Rangkaian Pembagi Arus**Error! Bookmark not defined.**

Gambar 4.13 Grafik Pengukuran Arus pada Rangkaian Elektronik**Error! Bookmark not defined.**

Gambar 4.14 Pengambilan Data PWM dengan Hasil Amplitudo 0,5 volt.....**Error! Bookmark not defined.**

Gambar 4.15 Pengambilan Data PWM dengan Hasil Amplitudo 1 volt**Error! Bookmark not defined.**

DAFTAR TABEL

HALAMAN

Tabel 3.1 Perhitungan Torsi Motor <i>Skateboard</i>	Error! Bookmark not defined.
Tabel 3.2 Tabel Spesifikasi Motor DC <i>Geared</i>	Error! Bookmark not defined.
Tabel 3.3 Tabel Spesifikasi Motor DC MY1025	Error! Bookmark not defined.
Tabel 3.4 Tabel Kecepatan Sesuai dengan Ukuran Diameter Roda	Error! Bookmark not defined.
Tabel 4.1 Hasil Pengukuran Tegangan <i>Output</i> Baterai (<i>accu</i>)	Error! Bookmark not defined.
Tabel 4.2 Hasil Pengujian Sensor <i>Accelerometer</i>	Error! Bookmark not defined.
Tabel 4.3 Hasil Pengukuran RPM Tanpa Beban	Error! Bookmark not defined.
Tabel 4.4 Hasil Pengukuran RPM dan Kecepatan dengan Beban	Error! Bookmark not defined.
Tabel 4.5 Hasil Pengukuran Tegangan <i>Output Port D</i>	Error! Bookmark not defined.
Tabel 4.6 Hasil Pengukuran Tegangan Masukan Motor DC	Error! Bookmark not defined.
Tabel 4.7 Pengujian Arus Tanpa Beban dan dengan Sebesar ± 60 kg	Error! Bookmark not defined.
Tabel 4.8 Pengujian Jarak Komunikasi <i>Bluetooth</i>	Error! Bookmark not defined.
Tabel 4.9 Pengujian Aplikasi Menggunakan <i>Device</i> Lain	Error! Bookmark not defined.

-----Halaman ini sengaja dikosongkan-----

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi, *skateboard* dituntut untuk mampu berkembang dibidang kecanggihan. Dengan memasang beberapa komponen elektronika, *skateboard* menjadi kendaraan listrik modern di negara-negara maju. Mainan yang sering di asosiasikan dengan kalangan remaja ini, sekarang memiliki perspektif baru sebagai solusi sarana transportasi di kota – kota besar. *Skateboard* dijadikan kendaraan listrik pribadi karena ukurannya yang kecil, ringan, dan mudah dibawa. *Skateboard* listrik mampu menjawab permasalahan masyarakat kota yang memiliki mobilitas tinggi saat ini, diantaranya kemacetan, polusi, penghematan bahan bakar minyak, dan praktis.

Skateboard mulai dibuat dengan sistem pengatur kecepatan. Sistem kecepatan pada *skateboard* listrik menggunakan sistem pengendali dengan metode push button yang dimasukkan sebagai kontrol motor. *Skateboard* tidak dapat berjalan dalam waktu yang lama hanya mampu berjalan sekitar 5 – 8 menit. Selain itu kecepatan maksimal hanya 2,81 km/ jam dengan beban sebesar 50 kg. [1]

Melihat metode sebelumnya memiliki kekurangan yaitu kecepatan maksimal yang masih rendah serta pengoperasiannya yang kurang efektif. Oleh karena itu, Penggunaan *skateboard* dengan pengendali kecepatan menggunakan *smartphone* akan lebih memudahkan penggunaanya. *Skateboard* akan diberikan aplikasi yang *user friendly* untuk mengontrol kecepatan motor sesuai dengan yang dikehendaki. Aplikasi tersebut akan mengontrol kecepatan motor sesuai dengan data sensor *accelerometer* yang terdapat pada *smartphone*.

Pengontrolan kecepatan perantara *smartphone* lebih efektif karena mayoritas masyarakat telah menggunakan *smartphone* dikehidupan sehari - hari. Sehingga tidak perlu memiliki *device* lain lagi untuk menggunakan *skateboard* elektrik. Karena fungsi *smartphone* juga sebagai *remote control*. Pengguna *smartphone* hanya perlu memiliki aplikasinya saja tanpa harus membeli *remote control* khusus.

1.2 Permasalahan

Melihat latar belakang di atas, maka dirumuskan permasalahan antara lain :

1. *Skateboard* listrik tidak populer di Indonesia akibat desainnya yang tidak sesuai dengan tekstur jalan dan *spareparts* yang susah didapatkan.
2. Penggunaan *skateboard* listrik yang dirasa belum maksimal.

1.3 Batasan Masalah

Batasan masalah yang akan dibahas dalam pembuatan Tugas Akhir ini, yaitu:

1. Mikrokontroler pada rangkaian hanya mengatur kecepatan motor.
2. Alat ini hanya dapat mengatur gerakan maju dan mundur, untuk gerakan berbelok dilakukan secara manual.

1.4 Tujuan

Tujuan penulisan tugas akhir ini adalah:

1. Merancang dan merealisasikan *skateboard* listrik yang praktis, murah, dan cocok digunakan pada lajur bergelombang.
2. Memperoleh pemicu *skateboard* listrik yang canggih dan mudah didapatkan.

1.5 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan metodologi, yaitu, studi literatur, perencanaan dan pemodelan sistem, pembuatan *hardware*, perencanaan dan pembuatan *software*, uji coba dan analisa data, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap studi literatur akan dipelajari mengenai literatur yang sesuai dengan judul sebagai acuan untuk mengerjakan tahap selanjutnya. Literatur yang diperlukan meliputi cara kerja sensor *accelerometer*, kontrol kecepatan motor DC dan pemrograman pada mikrokontroler ATmega16 dengan CV AVR. Selanjutnya, mempelajari karakteristik motor yang akan digunakan. Selanjutnya pembuatan *hardware*, membuat rancangan *skateboard* elektronik menggunakan pemicu sensor *accelerometer* Android, dimulai pembuatan *hardware* dengan membuat mikrokontroler ATmega16, membuat perancangan *skateboard* sesuai dengan desain yang telah dibuat. Perencanaan dan pembuatan *software*, dimulai dari program untuk menjalankan *driver* motor dc sehingga motor dapat berfungsi dengan baik dengan sistem mekaniknya. Setelah dilakukan simulasi, pengujian alat bertujuan untuk memastikan bahwa

setiap komponen dari pembuatan alat dapat berfungsi sesuai dengan yang diinginkan. Data percobaan yang telah diperoleh selanjutnya akan dianalisis. Dari hasil analisis, akan ditarik kesimpulan dari penelitian yang telah dilakukan. Tahap akhir penelitian adalah penyusunan laporan penelitian.

1.6 Sistematika Laporan

Dalam penyusunan buku tugas akhir ini, pembahasan mengenai sistem alat yang di buat akan menjadi 5 bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Mendiskripsikan tentang latar belakang, perumusan masalah, batasan masalah, tujuan, sistematika penulisan, serta relevansi yang berkaitan dengan judul Tugas Akhir ini.

Bab II Teori Dasar

Berisi penjelasan mengenai dasar teori penunjang yang berkaitan dengan Tugas Akhir.

Bab III Perancangan dan Pembuatan Alat

Pembahasan secara detail tentang perancangan alat yang akan dibuat serta bahan dan alat yang akan digunakan selanjutnya pada tahap pembuatan alat. Perancangan alat dimulai dari perancangan mekanik setelah itu perancangan dan pembuatan elektriknya.

Bab IV Pengujian dan Analisa Data

Berisi data-data pengujian alat pada *hardware* dan *software* secara keseluruhan beserta analisisnya.

Bab V Penutup

Berisi kesimpulan yang di peroleh dari pembuatan tugas akhir ini serta saran yang membangun agar dapat mengembangkan alat selanjutnya.

1.7 Relevansi

Manfaat yang bisa diperoleh dari perancangan alat untuk Tugas Akhir adalah sebagai berikut :

1. Dapat mengembangkan *skateboard* menjadi kendaraan listrik pribadi.
2. Dapat mempermudah pengguna *skateboard* untuk mengontrol kecepatan sesuai keinginannya.
3. Mengatasi permasalahan kemacetan dan penghematan bahan bakar minyak. Serta lebih efisien dan menghemat waktu.

BAB II

TEORI DASAR

2.1 Tinjauan Pustaka

Ada dua metode yang pernah diusulkan untuk menyelesaikan masalah kontrol kecepatan pada *skateboard*. Pada [1] sistem kecepatan pada *skateboard* listrik menggunakan sistem pengendali dengan metode push button yang dimasukkan sebagai kontrol motor. *Skateboard* tidak dapat berjalan dalam waktu yang lama hanya mampu berjalan sekitar 5 – 8 menit. Selain itu kecepatan maksimal hanya 2,81 km/ jam dengan beban sebesar 50 kg.

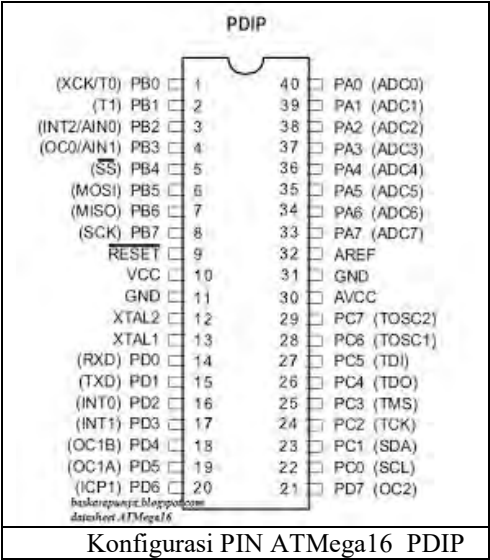
Pada [2] *Skateboard* menggunakan kontrol melalui kabel karena penting bahwa sinyal dari *remote control unit* dari satu pengguna sehingga tidak akan mengganggu unit remote control dari pengguna lain. Unit kontrol radio dilengkapi dengan sirkuit sehingga *unit remote control* tertentu hanya akan mengontrol satu controller.

Pada Tugas Akhir ini akan dilakukan perancangan kontrol kecepatan pada *skateboard* dengan menggunakan aplikasi pada *smartphone*. Penggunaan *remote control* kurang efisien, karena pengguna harus memiliki *remote control* untuk mengoperasikan *skateboard*. Namun, dengan *smartphone* masyarakat tidak perlu membeli remote kontrol namun hanya menginstall aplikasi. Sehingga lebih hemat serta efisien. Selain itu kontrol kecepatan *skateboard* dapat sesuai dengan yang diharapkan.

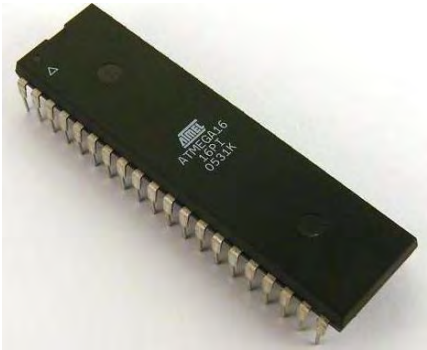
2.2 Mikrokontroler ATmega16

Mikrokontroler merupakan sistem komputer lengkap dalam bentuk *chip*. Pada mikrokontroler didalamnya terdapat ROM (*Random Only Memory*), RAM (*Random Acces Memory*), ADC (*Analog to Digital Converter*), DAC (*Digital to Analog Converter*), *input* dan *output* serta serial komunikasi berada dalam satu *chip*. Bentuk fisik dari ATmega16 seperti pada Gambar 2.2. Mikrokontroler ATmega16 memiliki pin *input/output* yang berjumlah 32 buah, yang terbagi menjadi 4 *port* yaitu *Port A*, *Port B*, *Port C* dan *Port D*. Konfigurasi pin yang dimiliki mikrokontroler ATmega16 adalah 40. Konfigurasi pin pada ATmega16 ditunjukkan pada Gambar 2.1. Bahasa pemrograman yang digunakan mikrokontroler yaitu bahasa C dan assembly. Terdapat perbedaan antara ATmega16 dan ATmega16L diantaranya yaitu, pada operasi tegangan

dan kecepatannya. Pada ATmega16L, bekerja pada tegangan 2,7 – 5,5 V sedangkan pada ATmega16 operasi tegangannya 4,5 – 5,5 V. Kecepatan pada ATmega16L sekitar 0 – 8 MHz, sedangkan ATmega16 adalah 0 – 16 MHz.[3]



Gambar 2.1 Konfigurasi Pin ATmega 16



Gambar 2.2 Bentuk Fisik ATmega16

2.3 Motor DC

Motor merupakan peralatan elektromagnetis yang mampu mengubah energi listrik menjadi energi mekanik. Ada beberapa jenis motor, namun yang digunakan yaitu motor dc. Motor dc atau sering disebut dengan motor arus searah merupakan jenis motor dc yang menggunakan tegangan searah sebagai sumber tegangannya. Pada motor dc ini terdapat dua bagian yaitu bagian strator dan rotor. Strator atau sering diketahui sebagai bagian motor yang tetap. Pada strator menghasilkan medan magnet yang dibangkitkan dari sebuah koil ataupun magnet permanen. Sedangkan rotor merupakan bagian motor dc yang berputar. Rotor terdiri dari sebuah koil dimana arus listrik akan mengalir. Prinsip kerja pada motor dc yaitu membalikkan fasa tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator, arus akan berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet.

Bagian atau komponen yang terdapat pada motor dc yaitu sebagai berikut :

- a. Badan Motor : bagian motor yang digunakan sebagai tempat untuk meneruskan fluks magnet yang dihasilkan oleh kutub – kutub magnet. Pada badan motor terdapat papan nama (*name plate*) yang berisi spesifikasi umum dari motor.
- b. Inti kutub magnet : bagian motor yang memiliki fungsi untuk mengalirkan arus listrik agar dapat terjadi proses elektromagnetik.
- c. Sikat-sikat : bagian motor yang berfungsi untuk menghubungkan aliran arus dari lilitan jangkar dengan sumber tegangan.
- d. *Commutator* : adalah komponen yang paling utama dalam motor DC. Kegunaannya adalah untuk mengalirkan arus antara dinamo dan sumber daya.
- e. Belitan jangkar : bagian yang terpenting pada mesin arus searah, berfungsi sebagai tempat munculnya tenaga putar pada motor.[4]



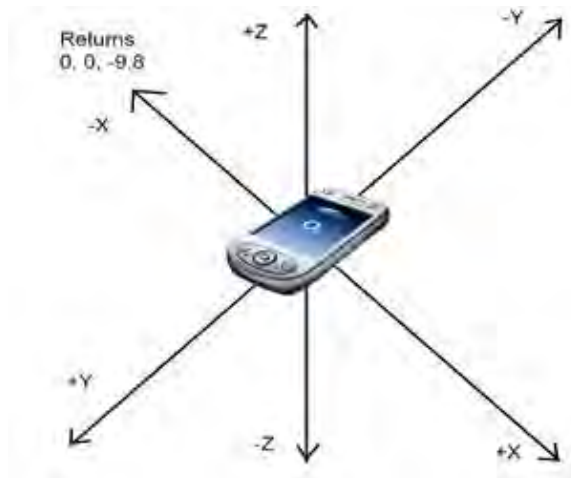
Gambar 2.3 Motor DC MY1025

Gambar 2.3 merupakan bentuk fisik dari motor dc yang digunakan pada tugas akhir ini. Spesifikasi motor dc yang digunakan pada tugas akhir ini yaitu motor dc. Motor dc ini memiliki torsi $\pm 85 - 100 \text{ kg.cm}$ atau sekitar $8,3 - 9,8 \text{ Nm}$. Daya yang dibutuhkan untuk menjalankan motor dc ini yaitu 250 watt. Sedangkan arus yang diperlukan yaitu 14 A.[4]

2.4 Sensor *Accelerometer*

Sensor *accelerometer* merupakan sebuah sensor yang dapat mengukur dan mendeteksi percepatan, mengukur getaran, atau mengukur percepatan gravitasi. Percepatan adalah bertambahnya suatu kecepatan dalam suatu rentang waktu. *Accelerometer* dapat digunakan untuk mengukur percepatan baik statis maupun dinamis. Pengukuran percepatan dinamis adalah pengukuran percepatan pada objek yang bergerak, sedangkan percepatan statis adalah pengukuran percepatan terhadap gravitasi bumi.

Letak sensor *accelerometer* ditanam didalan *smartphone*. Pada *smartphone* sensor ini digunakan untuk menentukan derajat kemiringan dari *smartphone*. Sensor ini juga dapat mengukur kemiringan suatu benda dikarenakan memiliki 3 sumbu yaitu x, y, dan z. Nilai x, y, dan z yang dihasilkan akan disesuaikan dengan perubahan posisi pada sumbu tertentu. Pada *smartphone* sensor ini dapat membaca pergerakan sehingga dapat mengubah tampilan layar dari posisi *landscape* ke *portrait* atau sebaliknya dengan cukup memiringkan badan *smartphone* secara otomatis.



Gambar 2.4 Sumbu pada Sensor *Accelerometer*

Gambar 2.4 merupakan ilustrasi pembagian sumbu x, y, z pada *trixial accelerometer*. Nilai x, y, z yang dihasilkan akan disesuaikan dengan perubahan posisi pada sumber tertentu. Aktivitas dari *user* yang berbeda akan menghasilkan nilai yang berbeda juga sesuai dengan cara *user* berinteraksi dengan *smartphone*.

- x : Jika *smartphone* digerakkan mengarah kebawah sesuai dengan sumbu maka x akan semakin berkurang sesuai percepatan.
- x : Jika *smartphone* digerakkan mengarah keatas sesuai dengan sumbu maka x akan semakin bertambah sesuai percepatan.
- y : Jika *smartphone* digerakkan mengarah kebawah sesuai dengan sumbu maka y akan semakin berkurang sesuai percepatan.
- y : Jika *smartphone* digerakkan mengarah keatas sesuai dengan sumbu maka y akan semakin bertambah sesuai percepatan.
- z : Jika *smartphone* digerakkan mengarah kebawah sesuai dengan sumbu maka z akan semakin berkurang sesuai percepatan.
- z : Jika *smartphone* digerakkan mengarah keatas sesuai dengan sumbu maka z akan semakin bertambah sesuai percepatan.[5]

2.5 HC – 05

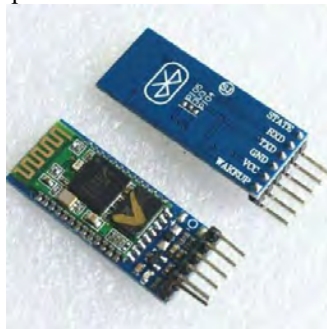
Bluetooth merupakan teknologi komunikasi *wireless* atau tanpa kabel yang digunakan untuk transimisi data. *Bluetooth* dapat bekerja pada pita frekuensi 2,4 Ghz dengan kecepatan *transfer* data kurang data

dari 1 Mbps atau sekitar 800 Kbps. *Bluetooth* ini digunakan untuk transmisi data antar *device*. Sebagai contoh untuk mengirim data antar *smartphone*, dari *smartphone* ke laptop. Pada tugas akhir ini komunikasi *bluetooth* digunakan untuk mengirimkan dan menerima data dari *smartphone* ke mikrokontroler. Jarak jangkauan pada *bluetooth* terbatas, yaitu sekitar 10 meter tanpa penghalang.

Kelebihan dari *bluetooth* adalah *bluetooth* dapat menembus dinding serta benda penghalang lain walaupun jarak transmisinya terbatas. *Bluetooth* juga tidak membutuhkan kabel ataupun kawat untuk mentransmisikan datanya.

Tidak hanya kelebihan namun, *bluetooth* juga memiliki kekurangan. Kekurangan dari *bluetooth* yaitu banyak mekanisme keamanan *bluetooth* yang harus diperhatikan untuk mencegah kegagalan pengiriman atau penerimaan data. Apabila dalam suatu ruangan terdapat lebih dari satu koneksi *bluetooth* maka akan menyulitkan pengguna untuk menemukan *receiver* yang diharapkan.

Pada tugas akhir ini menggunakan modul *bluetooth* HC-05. Modul HC-05 ini mudah digunakan untuk komunikasi serial *wireless* yang mengkonversi *port* serial ke *bluetooth*. HC-05 bekerja pada gelombang radio berfrekuensi 2,4 GHz. Ukuran modul *bluetooth* ini sekitar 12.7mm x 27mm. Bentuknya yang sederhana dan kecil ini dapat memudahkan dalam mendesain alat. Serta tidak perlu membutuhkan ruang yang luas untuk menempatkannya pada alat.



Gambar 2.5 Modul *Bluetooth* HC -05

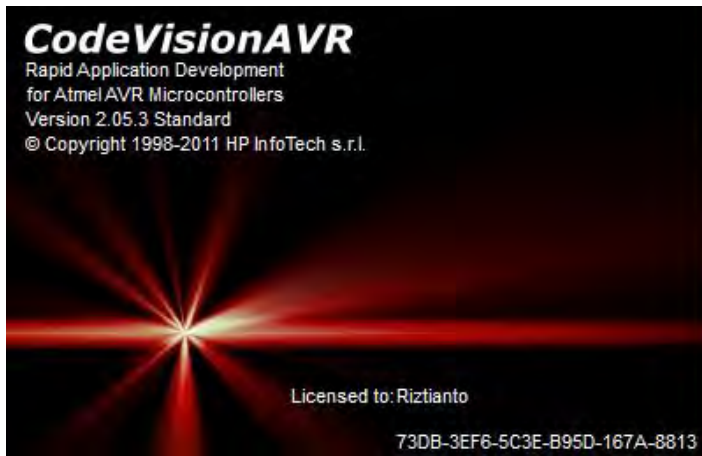
Module *bluetooth* HC-05 terdiri dari 6 pin. Namun yang digunakan hanya 4 pin, yaitu pin VCC,GND, TXD dan RXD. Konfigurasi dari pin HC-05 dapat dilihat pada Gambar 2.5. Berikut ini penjelasan masing – masing pin pada HC-05 :

1. VCC : pin yang berfungsi sebagai *input* tegangan. Pin ini dihubungkan pada sumber tegangan 5V
2. GND : pin ini berfungsi sebagai *ground*. Hubungkan pin ini dengan *ground* sumber tegangan
3. TX : pin yang berfungsi untuk mengirimkan data dari modul ke mikrokontroler.
4. RX : pin yang berfungsi untuk menerima data yang dikirim ke modul HC-05.
5. EN : pin berfungsi mengaktifkan mode *AT Command Setup* pada HC-05.
6. State : pin yang berfungsi memberikan informasi untuk memastikan modul dan perangkat lain terkoneksi dengan baik.[6]

2.6 Codevision AVR

Peralatan *hardware* mikrokontroler tidak dapat dijalankan tanpa adanya *software*. *Software* berfungsi untuk memasukkan perintah yang akan dieksekusi agar keluarannya sesuai dengan yang diharapkan. Pada tugas akhir ini menggunakan *software* yaitu CV AVR. *Software* CV AVR atau Codevision AVR adalah salah satu jenis perangkat lunak yang dapat digunakan untuk memprogram semua jenis mikrokontroler dari keluarga bahasa C. *Software* CV AVR menggunakan bahasa pemrograman C. Tampilan awal dari *software* CV AVR ditunjukkan pada Gambar 2.6. Code Vision AVR dapat dijalankan pada operasi windows 95, 98, Me, NT4, 200, dan XP.[7] Semua instruksi yang dapat diimplementasikan dengan fungsi-fungsi berikut :

1. Mengatur *set-up* akses memori eksternal
2. Mengidentifikasi sumber *reset chip*
3. Inisialisai *Port Input/Output*
4. Inisialisasi interupsi eksternal
5. Inisialisasi *Timer/Counter*
6. Inisialisasi antarmuka SPI
7. Inisialisasi ADC
8. Inisialisasi modul LCD.
9. Inisialisai USART



Gambar 2.6 CodeVision AVR

2.7 Baterai (Accu)

Perangkat elektronik diberikan catu daya arus searah DC (*direct current*) yang stabil agar dapat bekerja dengan baik. Baterai atau accu digunakan sebagai sumber catu daya DC. Baterai merupakan komponen yang dapat menyimpan muatan listrik searah berupa tegangan. Baterai memiliki terminal positif atau katoda, dan terminal negatif atau anoda. Terminal positif adalah kutub yang memiliki energi potensial yang lebih tinggi daripada kutub yang lain. Terminal negatif adalah sumber elektroda yang apabila dihubungkan dengan rangkaian lain akan mengalirkan energi ke rangkaian tersebut. Terdapat dua jenis baterai *accu* yaitu *accu* basah dan *accu* kering. Perbedaan antara keduanya yaitu *accu* basah masih perlu diberi air *accu* yang dikenal *accu zuur*. Sedangkan jenis *accu* kering merupakan jenis aki yang tidak memakai cairan, mirip seperti baterai telepon seluler. *Accu* ini tahan terhadap getaran dan suhu rendah.

Jenis aki kering yang digunakan yaitu VRLA. *Valve Regulated Lead Acid* (VRLA) yaitu baterai yang bersifat tertutup (*sealed*), sehingga penguapan yang dikeluarkan sangat kecil (rekombinasi) maka tidak memerlukan penambahan cairan elektrolit selama masa pemakaian baterai tersebut. Proses penguapan pada baterai diatur oleh bagian yang disebut *valve* (katup). Kelebihan VRLA yaitu baterai ini lebih aman, bebas perawatan serta dapat dipasang dalam kondisi apapun. Kapasitas

baterai yang digunakan yaitu 7,5 Ah 12 volt sebanyak dua buah. Kapasitas baterai menunjukkan kemampuan baterai untuk mengeluarkan arus (*discharging*) selama waktu tertentu, dinyatakan dalam Ah (Ampere – hour). Ini dapat diartikan sebuah baterai dapat memberikan arus yang kecil untuk waktu yang lama atau arus yang besar untuk waktu yang pendek. Pada saat baterai diisi terjadilah penimbunan muatan listrik.

Baterai (*accu*) berfungsi untuk memasok energi listrik kepada satu atau lebih beban listrik. Fungsi dari *accu* yaitu sebagai sumber tenaga. Jika baterai tidak bekerja dengan baik maka akan mempengaruhi kinerja *hardware* yang lain. Pada tugas akhir ini menggunakan *accu* kering dengan kapasitas 24V. Catu daya yang dibutuhkan yaitu 24V, 12 dan 5V, maka membutuhkan *regulator*. *Voltage regulator* berfungsi sebagai *filter* tegangan agar sesuai dengan keinginan.[8]

2.8 IC LM 2596

IC LM2596 merupakan sirkuit terpadu atau *integrated circuit* yang berfungsi sebagai *step-down DC converter* dengan maksimal *current rating* 3A. Terdapat dua jenis pada IC ini yaitu *adjustable*, dimana tegangan keluarannya dapat diatur, dan *fixed voltage output*, dimana tegangan keluarannya sudah tetap.



Gambar 2.7 Pin Konfigurasi IC LM2596

Konfigurasi pin pada IC LM2596 ditunjukkan pada Gambar 2.7. IC LM2596 bekerja pada switching frekuensi 150 kHz. Tegangan keluaran pada IC ini yaitu 5V sedangkan untuk tegangan masukan maksimum yaitu 40V. IC ini disebut *switching regulator* yaitu *regulator* yang bekerja dengan mengambil sedikit demi sedikit energi dari sumber tegangan *input* kemudian memindahkan energi tersebut ke *output*. [9]

2.9 Driver Motor

Motor dc digunakan untuk menggerakkan roda dari *skateboard*. Namun, pada motor dc sendiri tidak dapat dikendalikan langsung oleh keluaran dari mikrokontroler. Tegangan mikrokontroler hanya sebesar 5 volt, sedangkan motor dc yang digunakan pada *skateboard* membutuhkan tegangan sebesar 24 volt agar dapat bekerja dengan optimal. Rangkaian pengendali yang digunakan pada Tugas Akhir ini adalah rangkaian *driver* motor dengan IC BTS7960. Pada *driver* motor dc ini dapat mengeluarkan arus hingga 43A. Rangkaian *driver* motor ini digunakan untuk mengatur putaran motor dc dengan menggunakan metode PWM. PWM (*Pulse Width Modulation*) merupakan suatu teknik pengaturan kecepatan putaran motor dengan cara mengatur lebar pulsa sinyal dalam satu periode. Perputaran motor dc ini diatur dengan merubah polaritas dari tegangan yang bekerja pada motor dc.[10] Bentuk fisik dari *driver* motor dc BTS7960 seperti pada Gambar 2.8.



Gambar 2.8 Driver Motor BTS7960

Pin konfigurasi dari *driver* 43A H-Bridge Drive PWM ini dapat dilihat pada Gambar 2.9:



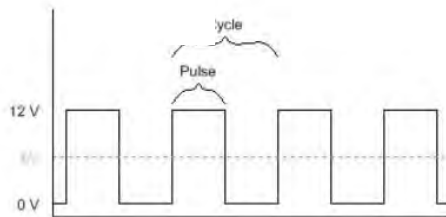
Gambar 2.9 Pin Konfigurasi Driver Motor BTS7960

1. RPWM : *forward level* atau sinyal PWM, aktif *high*
2. LPWM : *invers level* atau sinyal PWM, aktif *high*
3. R_EN : *forward drive enable input, high enable*

4. L_EN : *reverse drive enable input, high enable*
5. R_IS : *forward drive, current alarm output*
6. L_IS : *reverse drive, current alarm output*
7. VCC : *power input 5V*
8. GND : *Terminal ground*

2.10 PWM (Pulse Width Modulation)

Pulse Width Modulation adalah sebuah teknik modulasi dengan cara merubah lebar pulsa. PWM (*Pulse Width Modulation*) merupakan suatu teknik pengaturan kecepatan putaran motor dengan cara mengatur lebar pulsa sinyal dalam satu periode. Teknik PWM pada tugas akhir ini digunakan pada *driver* motor DC. PWM pada tugas akhir ini digunakan untuk mengatur *duty cycle* dengan mudah. *Duty cycle* adalah waktu yang dibutuhkan pada saat gelombang pulsa ada pada kondisi *high* atau aktif (dinyatakan dengan logika “1”). Gambaran mengenai PWM ini dapat kita amati pada Gambar 2.10.



Gambar 2.10 *Pulse Width Modulation*

Duty Cycle dinyatakan dalam bentuk presentase, cara untuk menentukan besar *duty cycle* adalah seperti pada persamaan 2.1.

$$D = \frac{t_{pulse}}{t_{cycle}} \times 100\% \dots \dots \dots (2.1)$$

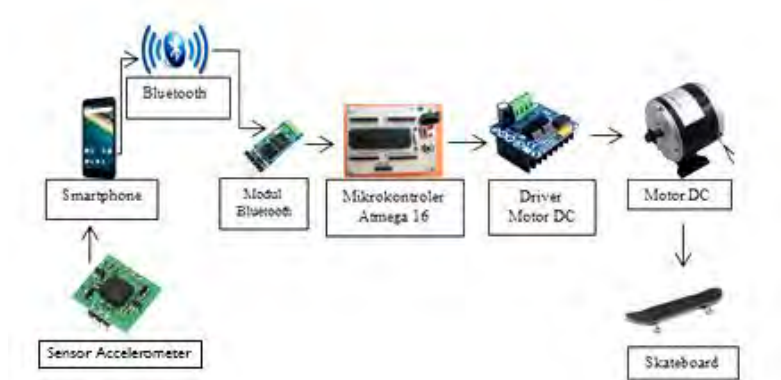
Dimana D adalah *duty cycle*, t_{pulse} adalah waktu dimana pulsa ada pada kondisi aktif, dan t_{cycle} adalah waktu untuk satu gelombang pulsa (satu puncak dan satu lembah).[7]

-----Halaman ini sengaja dikosongkan-----

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini membahas tentang tahapan yang dilakukan dalam perencanaan dan pembuatan alat pada tugas akhir. Permulaan dari bab ini akan dijelaskan mengenai blok fungsional sistem secara keseluruhan, setelah itu perancangan perangkat keras dan perancangan *software*. Berikut ini merupakan perancangan sistem secara keseluruhan dari tugas akhir :



Gambar 1.1 Perancangan Sistem Secara keseluruhan

Pada Gambar 3.1 merupakan sistem dari keseluruhan kerja alat pada Tugas Akhir ini. Sensor *accelerometer* pada *smartphone* berfungsi sebagai pemberi data yang akan diproses oleh aplikasi sebagai pengontrol *skateboard*. Data *accelerometer* yang sudah diproses oleh aplikasi ini akan dikirim ke mikrokontroler dengan perantara *bluetooth* sebagai media transmisi. Sebelumnya *bluetooth smartphone* yang digunakan *smartphone* harus terkoneksi dengan *bluetooth skateboard* yaitu HC-05. Selanjutnya data tersebut akan dikirimkan melalui media transmisi *bluetooth* yang telah terhubung dengan mikrokontroler ATmega16.

Data yang diterima *bluetooth* dari *smartphone* akan diproses oleh mikrokontroler ATmega16. Data tersebut akan dikirim ke *driver* motor

sebagai *input* motor, sehingga kecepatan motor dapat dikendalikan sesuai dengan yang dikehendaki pengguna *skateboard*. Pada *smartphone* menggunakan sensor *accelerometer* sebagai input data. Jika *bluetooth* tidak terkoneksi dengan baik maka proses pengiriman data akan mengalami kegagalan. Oleh karena itu perlu dilakukan pengecekan diawal pengoperasian alat. Untuk kecepatan motor diatur sedemikian rupa sehingga sesuai dengan keluaran yang diharapkan.

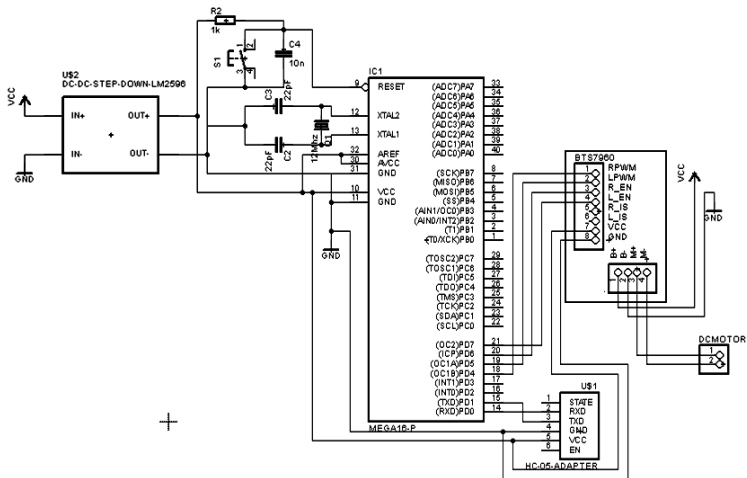
Dalam pembuatan tugas akhir ini diperlukan komponen pokok untuk merancang alat yang sesuai dengan sesuai yang diinginkan :

1. Mikrokontroler, mikrokontroler yang digunakan adalah mikrokontroler ATmega 16. Mikrokontroler ini digunakan untuk menyimpan program sensor juga sebagai pusat kendali sistem sesuai dengan pengaturan yang diinginkan dalam Tugas Akhir ini.
2. Sensor *Accelerometer*, berfungsi sebagai sensor yang mendeteksi dan mengukur getaran, ataupun untuk mengukur percepatan akibat gravitasi bumi.
3. *Smartphone*, berfungsi sebagai alat yang digunakan untuk mengirimkan data sensor *accelerometer*. Sensor *accelerometer* terdapat pada *smartphone* ini akan memberikan data yang terbaca oleh aplikasi.
4. Motor DC, berfungsi sebagai alat yang dikontrol kecepatannya.
5. Modul *Bluetooth* (HC-05), berfungsi untuk mengirimkan dan menerima data antara *smartphone* dengan mikrokontroler.
6. Catu Daya, digunakan untuk memberikan sumber tegangan untuk beberapa komponen elektronik. *Supply* tegangan yang dibutuhkan yaitu 5 volt, 12 volt dan 24volt.

1.1 Perancangan *Hardware*

Perancangan *hardware* dilakukan dengan merancang rangkaian – rangkaian elektronika dan rancangan mekanik. Perancangan *hardware* ini meliputi :

1. Rangkaian Mikrokontroler ATmega 16
2. Rangkaian *Driver* Motor DC
3. Rangkaian Catu daya
4. Rangkaian *Regulator*



Gambar 1.2 Skema Rangkaian Secara Keseluruhan

Langkah pertama untuk membuat rangkaian yaitu membuat skema. Skema rangkaian ini berfungsi untuk mengetahui rangkaian yang akan kita buat dapat dijalankan dengan baik atau tidak. Hal ini dapat dilakukan dengan mensimulasi rangkaian pada *software* simulasi. Simulasi ini bermaksud dapat membantu untuk mengetahui rangkaian yang akan dibuat dapat berjalan dengan baik sebelum dilakukan pembuatan rangkaian dengan komponen yang nyata. Ini dapat membantu menurunkan tingkat kegagalan pada rangkaian.

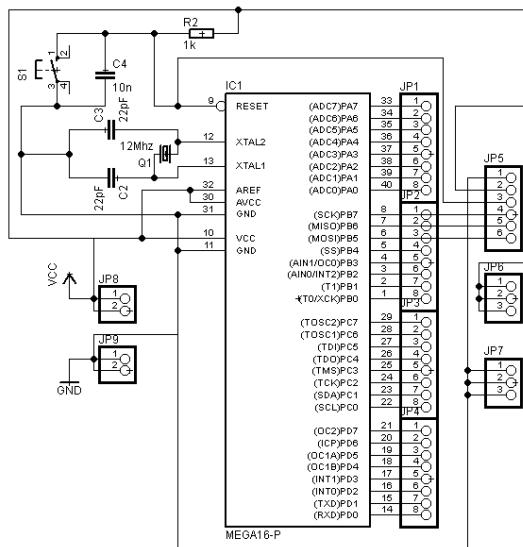
1.1.1 Perancangan Mikrokontroler ATmega 16

Sistem Minimum Mikrokontroler merupakan sebuah rangkaian paling sederhana dari sebuah mikrokontroler agar IC mikrokontroler tersebut dapat beroperasi dan diprogram. Rangkaian Mikrokontroler AVR ATmega 16 ini digunakan sebagai kontroler dari seluruh aktivitas yang dikerjakan oleh alat.

ATmega 16 terdiri dari 40 pin, terdapat *port A*, *port B*, *port C* dan *port D*. Ada juga pin MISO, MOSI, SCK beserta reset, VCC, dan GND yang dapat langsung dihubungkan ke *downloader* atau USB ASP. Mikrokontroler ATmega 16 digunakan untuk mengontrol seluruh

aktivitas dari sistem alat ini. Fungsi dari mikrokontroler ini adalah untuk membaca nilai sensor dari *smartphone*. Untuk membuat Minimum Sistem ATmega 16 berikut ini komponen yang dibutuhkan :

1. IC ATmega 16
2. Socket ATmega 16
3. PCB
4. Crystall Oscillator 16MHz
5. Kapasitor keramik 22pF (2 buah)
6. Kapasitor keramik 10n
7. Resistor 1k (2buah)
8. Push button
9. Pin Header Male/Female
10. DC Jack
11. Dioda 1N4004 (1A)
12. Kapasitor Elektrolit 2200uF/16V
13. Kapasitor Milar 100n
14. LED 3mm
15. LM7805



Gambar 1.3 Rangkaian Mikrokontroler ATmega16

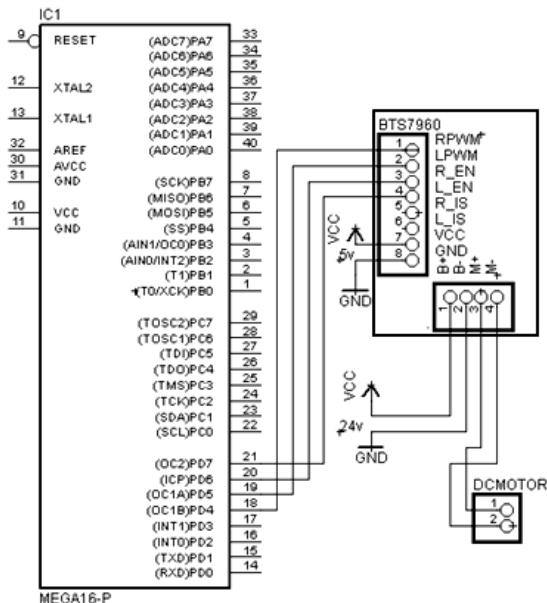
1.1.2 Rangkaian *Driver* Motor DC

Motor DC digunakan untuk menggerakkan roda dari *skateboard*. Namun, pada motor dc sendiri tidak dapat dikendalikan langsung oleh keluaran dari mikrokontroler. Tegangan mikrokontroler hanya sebesar 5 volt, sedangkan motor dc yang digunakan pada *skateboard* membutuhkan tegangan sebesar 24 volt agar dapat bekerja dengan optimal. Rangkaian pengendali yang digunakan pada Tugas Akhir ini adalah rangkaian *driver* motor dengan IC BTS7960.

Pada *driver* motor DC ini dapat mengeluarkan arus hingga 43A, dengan memiliki fungsi PWM. Penggunaan *driver* motor ini dikarenakan ketersediaan IC yang sulit di jangkau sehingga beralih menggunakan *driver* motor yang sudah berbentuk modul. Selain itu, arus maksimal yang mengalir pada motor dc yaitu 14 A. Sedangkan arus yang dibutuhkan untuk *start* awal motor yaitu 42 A sekitar tiga kali lipat dari arus yang mengalir pada motor dc.

Wiring dan rangkaian antara *driver* motor dan mikrokontroler. *Driver* motor berfungsi sebagai pengontrol motor dc. Berikut ini adalah wiring *driver* motor dc dengan mikrokontroler pada Gambar 3.4.

1. Port PD 7 disambungkan dengan kaki L_EN pada *driver* motor.
2. Port PD 6 disambungkan dengan kaki R_EN pada *driver* motor.
3. Port PD 5 disambungkan dengan kaki LPWM pada *driver* motor.
4. Port PD 4 disambungkan dengan kaki RPWM pada *driver* motor.
5. Kaki VCC pada *driver* motor disambungkan dengan kaki VCC pada HC-05.
6. Kaki GND pada *driver* motor disambungkan dengan kaki GND pada HC-05.
7. Sambungkan motor dc dengan *pin head* yang tersedia pada *driver* motor.
8. Sambungkan dua kaki *pin head* yang tersedia pada *driver* motor dengan VCC dan GND.

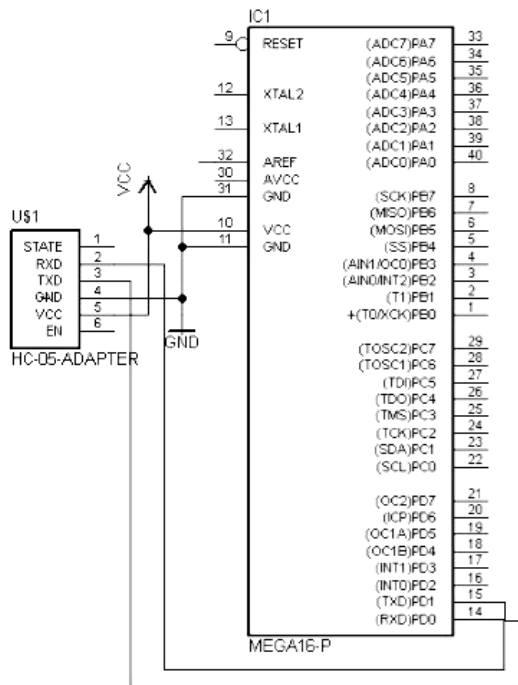


Gambar 1.4 Skema Rangkaian *Driver Motor* BTS7960 dengan Mikrokontroler

1.1.3 Rangkaian HC-05

Bluetooth HC-05 merupakan transmisi data wireless yang berfungsi sebagai penerima dan pengirim data. *Wiring* HC-05 dijelaskan pada Gambar 3.5.

1. Pin TXD disambungkan pada *port* PD 1 pada mikrokontroler. Pin TXD berfungsi sebagai pengirim data dari modul ke mikrokontroler.
2. Pin RXD disambungkan pada *port* PD 0 pada mikrokontroler. Pin RXD berfungsi sebagai menerima data dari mikrokontroler ke modul
3. Pin VCC disambungkan pada *port* VCC pada mikrokontroler.
4. Pin GND disambungkan pada *port* GND pada mikrokontroler.



Gambar 1.5 Skema rangkaian HC-05 dengan mikrokontroler

1.1.4 Rangkaian Catu Daya

Catu daya merupakan bagian penting dalam sebuah rangkaian elektronika. Fungsi dari catu daya yaitu sebagai sumber tenaga. Catu daya mensuplay arus dan tegangan ke seluruh rangkaian yang ada. Komponen ini akan mencatu tegangan sesuai dengan tegangan yang diperlukan oleh rangkaian elektronika.

Rangkaian terminal baterai digunakan untuk menyediakan tegangan DC yang berfungsi untuk seluruh kebutuhan sumber energi pada alat yang akan digunakan. Pada Tugas akhir ini menggunakan catu daya dari accu kering dengan tegangan 12 volt dan kapasitas arusnya adalah 7,5Ah. Spesifikasi *constant voltage charge*, dengan penggunaan *voltage* antara 14,4 – 15V. Initial current kurang dari 3A. Karena suplai yang dibutuhkan untuk menggerakkan motor dc sebesar 24 volt maka digunakan 2 buah *accu* kering yang disusun secara seri agar memenuhi

catu daya yang dibutuhkan. Penggunaan *accu* kering dirasa lebih efektif dikarenakan penggunaan baterai lipo (litium polimer) yang harganya terlalu mahal serta spesifikasi baterai yang sesuai dengan kebutuhan jarang dijual dipasaran. Selain itu kapasitas baterai lipo yang kecil serta baterai lipo yang tidak tahan lama karena akan mudah sekali untuk mengembung dalam beberapa kali pengisian baterai.

Catu daya yang dibutuhkan pada Tugas Akhir ini adalah 5 volt, 12 volt dan 24 volt. Untuk mendapatkan tegangan sebesar 5 volt menggunakan IC *Regulator* LM2596. Keluaran *power supply* 5 volt digunakan untuk masukan mikrokontroler, *driver* motor dc dan masukan untuk *bluetooth* HC-05. Catu daya 24 volt dihubungkan pada motor dc sedangkan 12 volt dihubungkan pada *regulator*.



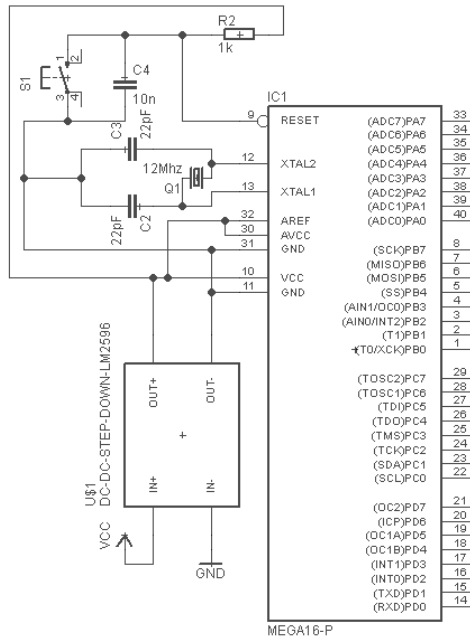
Gambar 1.6 Baterai 12 volt

Gamabr 3.6 merupakan skema rangkaian catu daya 12 v dan 24 volt. Perhitungan catu daya yang dibutuhkan voltage sebesar 24 V serta daya yang digunakan 250 watt. Lama waktu penggunaan yaitu sekitar 21,8 menit.

1.1.5 Regulator

Regulator merupakan rangkaian regulasi atau pengatur tegangan keluaran dari sebuah catu daya agar efek dari naik atau turunnya tegangan jala-jala tidak mempengaruhi tegangan catu daya sehingga menjadi stabil. *Regulator* memiliki fungsi untuk mempertahankan dan memastikan tegangan pada level tertentu secara otomatis. Rangkaian 12V DC – 5V DC converter berfungsi untuk mengubah keluaran tegangan sebesar 12 V dari aki menjadi tegangan 5V yang berfungsi untuk mencatu sebagian rangkaian sistem yang lain. Rangkaian ini terdiri dari rangkaian utama yaitu IC LM2596. LM 2596 merupakan IC *regulator* tegangan yang menggunakan metode pensaklaran. *Regulator*

tegangan dengan metode pensaklaran pada dasarnya adalah rangkaian konverter DC ke DC yang dilengkapi dengan sistem umpan balik. Pengaturan tegangan keluaran pada *regulator* ini dilakukan dengan mengubah duty cycle dari komponen IC *regulator* tersebut.



Gambar 1.7 Skema Rangkaian *Regulator* LM2596 dengan Mikrokontroler

Berikut ini penjelasan skema wiring rangkaian *regulator* dengan mikrokontroler pada Gambar 3.7.

1. Kaki in+ disambungkan dengan kaki VCC pada mikrokontroler
2. Kaki in- disambungkan dengan kaki GND pada mikrokontroler

1.2 Perhitungan Torsi Roda

Dalam perancangan *skateboard* elektrik diperlukan perhitungan torsi yang dibutuhkan roda untuk bergerak. Tujuan dari perhitungan torsi ini yaitu agar didapatkan motor DC dengan torsi yang tepat supaya dapat menggerakkan *skateboard* dengan spesifikasi yang telah ditentukan. Perhitungan torsi ini sangat dibutuhkan agar *skateboard*

dapat berjalan dengan baik dan tidak merusak komponen lain seperti motor DC.

Beberapa spesifikasi untuk menggerakkan *skateboard* telah ditentukan pada tabel 3.1, sehingga dapat diperoleh nilai-nilai *resistance* atau gaya yang dibutuhkan pada roda untuk dapat bergerak atau berputar pada kondisi yang ditentukan dengan beban orang maksimum. Resistansi tersebut antara lain adalah resistansi *rolling*, resistansi *grade*, dan resistansi akselerasi.

Tabel 1.1 Perhitungan Torsi Motor *Skateboard*

PERHITUNGAN TORSI		
Berat <i>skateboard</i>		10 kg
Berat maksimum orang		60 kg
Kemiringan maksimum		15°
Kecepatan maksimum		10 km/jam
Waktu start hingga kecepatan maksimum		10s
Gaya gravitasi (g)		9,8 m/s ²
Koefisien Gesekan aspal		0,022
Koefisien Bahan roda		0,4
<i>Rolling Resistance (RR)</i>		15,092N
<i>Grade Resistance (GR)</i>		176,988 N
<i>Acceleration Force (FA)</i>		19,44 N
TTE		153,6216 N
Torsi :		
TW		21,46 Nm
<i>Reality Check :</i>		
MTT	1 <i>wheel</i>	8,7122 Nm
	Total	34,8488 Nm
MTT > TW		

Penjelasan dan perhitungan torsi yang dibutuhkan oleh motor adalah sebagai berikut:

1. *Gross vehicle weight*

Gross vehicle weight (GVW) adalah berat total kendaraan ditambah berat muatan dalam satuan Newton.

$$\begin{aligned} GVW &= (\text{Berat kursi roda} + \text{berat maksimum orang}) \times g \\ &= (10 + 60) \times 9,8 \\ &= 686 \text{ N} \end{aligned}$$

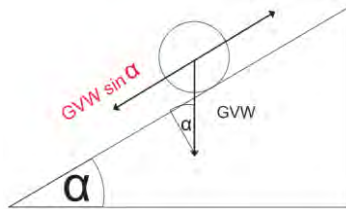
2. *Rolling resistance*

Rolling resistance merupakan gaya yang dibutuhkan roda untuk bergerak pada suatu permukaan tertentu. Pada perhitungannya, digunakan friksi permukaan dari jalanan beraspal dengan asumsi kondisi yang paling buruk, yaitu 0,022.

$$\begin{aligned} RR &= GVW \times \text{Koef. gesekan aspal} \\ &= 686 \times 0,022 \\ &= 15,092 \text{ N} \end{aligned}$$

3. *Grade resistance*

Grade resistance merupakan gaya yang dibutuhkan roda untuk tetap berjalan dalam kondisi jalanan yang miring atau menanjak dengan sudut jalan hingga 10 derajat. Sehingga beban yang harus ditanggung *skateboard* adalah berat total dikali sinus dari 10 derajat. Perhitungan *skateboard* di benda miring dapat diasumsikan seperti pada 3.8.



Gambar 1.8 Pemodelan Beban di Bidang Miring

$$\begin{aligned} GR &= GVW \times \sin \alpha \\ &= 686 \times \sin 10^\circ \\ &= 686 \times 0.1736 \\ &= 119,0896 \text{ N} \end{aligned}$$

4. Acceleration force

Acceleration force yaitu torsi yang dibutuhkan agar *skateboard* dapat mencapai kecepatan tertentu dalam waktu tertentu. Kecepatan adalah *skateboard* dapat mencapai kecepatan maksimum 10 km/jam dengan jangka waktu 10 detik dari keadaan diam. Dari spesifikasi tersebut didapatkan nilai *acceleration force* sebagai berikut:

$$\begin{aligned}
 FA &= \frac{GVW \times V_{\max}}{g \times t_{\max}} \\
 &= \frac{686 \times 10 \text{ km / jam}}{9,8 \times 10} \\
 &= \frac{686 \times 10 \frac{1000}{3600}}{9,8 \times 10} \\
 &= 19,44N
 \end{aligned}$$

5. Total tractive efforts

Total tractive efforts adalah kondisi terburuk pada saat semua spesifikasi di atas terpenuhi. Sehingga semua gaya yang diperlukan untuk masing – masing spesifikasi dijumlahkan menjadi 1 gaya total yang dibutuhkan. Sehingga pada kondisi jalanan menanjak pada sudut 15 derajat di jalanan beraspal dengan kecepatan maksimum 10 km/jam dan dengan waktu 10 detik untuk mencapai kecepatan maksimum, dibutuhkan gaya total dengan perhitungan sebagai berikut :

$$\begin{aligned}
 TTE &= RR + GR + FA \\
 &= 15,092 + 119,0896 + 19,44 \\
 &= 153,6216 N
 \end{aligned}$$

6. Wheel Motor Torque

Torque Wheel adalah torsi yang dibutuhkan untuk menggerakkan roda. Setelah diketahui gaya yang dibutuhkan, torsi total didapatkan dari perkalian gaya dengan r atau jarak tempuh. Dalam perhitungannya dibutuhkan nilai R_f dan R_{wheel} . R_f adalah total kondisi torsi motor dengan nilai 10 % torsi. Namun pada perhitungannya nilai R_f diasumsikan menjadi 11% yang merupakan penambahan 1 % karena ada kondisi hilangnya torsi karena pemakaian *gear*, *belt*, dan *shaft* yang digunakan. Sedangkan R_{wheel} adalah diameter roda *skateboard*. Torsi total ini tidak

terpengaruh oleh banyaknya roda yang digunakan. Torsi total ini akan memiliki nilai tetap berapapun jumlah roda yang digunakan. Perhitungan torsi total sebagai berikut :

$$\begin{aligned} TW &= TTE \times R_{wheel} \times R_f \\ &= 153,6216 \times 0,127 \times 1,1 \\ &= 21,46 \end{aligned}$$

7. Reality Check

MTT merupakan singkatan *maximum tractive force*. MTT mengindikasikan torsi maksimum yang dapat diaplikasikan pada *skateboard* agar ban tidak selip. Nilai torsi total ini harus lebih besar dari torsi yang dibutuhkan untuk memenuhi spesifikasi di atas, agar keamanan masih terjamin dan ban tidak akan selip.

Untuk tiap roda :

$$\begin{aligned} MTT &= Gaya \text{ tiap roda} \times Koef. \text{ bahan roda} \times Diameter \text{ roda} \\ &= (686 \div 4) \times 0,4 \times 0,127 \\ &= 8,7122 \end{aligned}$$

Maka total MTT untuk keseluruhan roda yaitu:

$$MTT = 8,7122 \times 4 = 34,8488$$

Karena $MTT > TW \Leftrightarrow 34,8488 > 21,46$ maka ban tidak akan slip.

Dengan rasio gear pada *skateboard* sebesar 1:2 maka torsi yang dibutuhkan untuk motor adalah

$$\frac{21,46}{2} = 10,73 Nm$$

1.3 Perancangan Motor DC

Pada alat ini diperlukan motor dc untuk menggerakkan roda skateboard. Motor dc yang digunakan pertama yaitu motor dc starter. Motor dc ini memiliki kecepatan hingga 500 RPM. Motor dc ini sering diaplikasikan pada sepeda motor delain itu juga diaplikasikan untuk roda pada kursi roda elektrik. Pada Tabel 3.2 menunjukkan spesifikasi dari motor dc yang digunakan pertama kali.

Tabel 1.2 Tabel Spesifikasi Motor DC *Geared*

Spesifikasi Motor DC	
V suplai	12 V (DC)
Arus	4 A
Speed	500 RPM
Torsi	100 kg.cm
Ukuran body	Panjang = 12,5 cm ; diameter = 5 cm
Ukuran shaft	Panjang = 1,5 cm ; diameter 1,5 cm
Berat	930 gram

Penggunaan motor dc dengan spesifikasi pada Tabel 3.2 dirasa belum memenuhi kebutuhan dari tugas akhir ini sehingga diperlukan adanya pergantian motor dc. Motor dc yang digunakan yaitu motor dc MY1025 dengan daya 250 watt serta motor dc ini memiliki kecepatan hingga 2750 RPM. Lebih lengkapnya dapat dilihat pada Tabel 3.3

Tabel 1.3 Tabel Spesifikasi Motor DC MY1025

Spesifikasi Motor DC	
V suplai	24 V (DC)
Arus	14 A
Speed	2750 RPM
Torsi	85 - 100 kg.cm
Daya	250 watt
Berat	1,9 kg

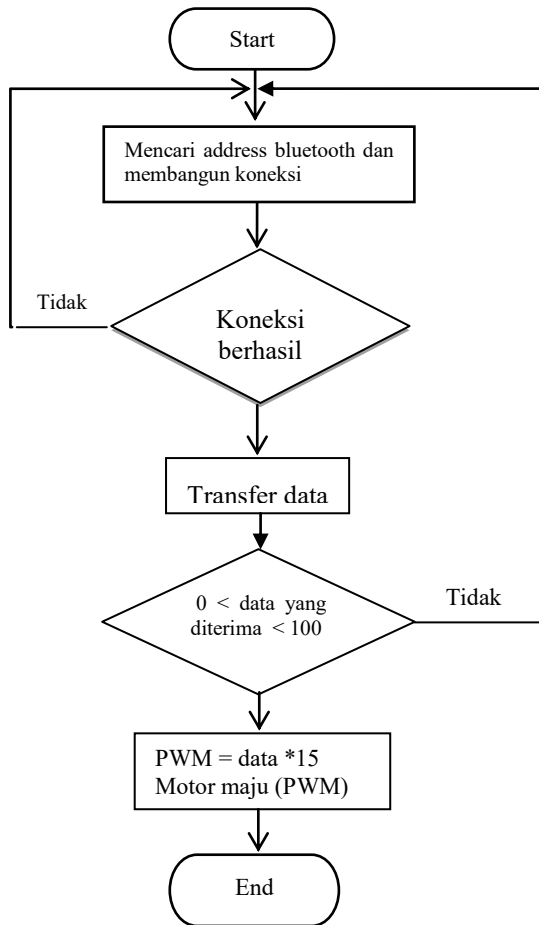
1.4 Perancangan *Software*

Perancangan *software* dilakukan dengan merancang program untuk dapat menggerakkan motor dc dengan baik. Perancangan *software* ini meliputi :

1. Perancangan program pada *software* CV AVR
2. Perancangan program untuk aplikasi *smartphone*

1.4.1 Perancangan Program pada *Software CV AVR*

Pada tahap perancangan program di CodeVision AVR, program yang akan dibuat meliputi program pengatur kecepatan motor dc, dan pengiriman data melalui *bluetooth*. Sebelum melakukan perancangan program, terlebih dahulu membuat *flowchart* atau diagram alir dari sistem secara keseluruhan. Berikut ini merupakan *flowchart* sistem secara keseluruhan pada Gambar 3.9 :



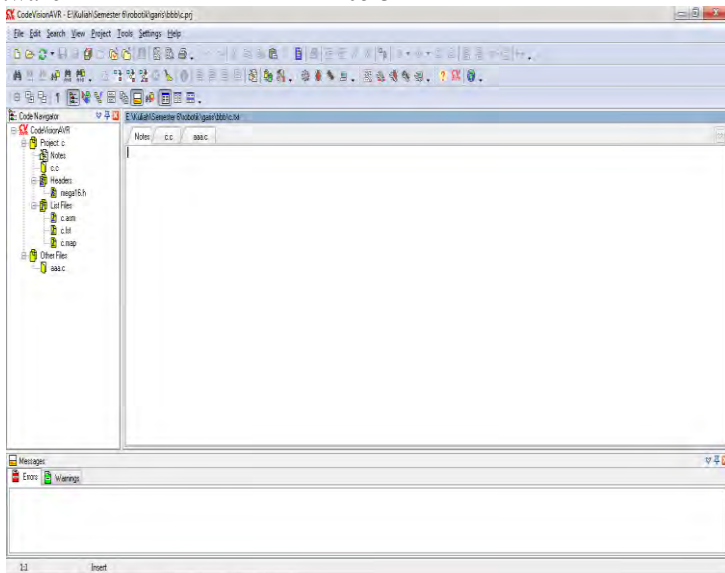
Gambar 1.9 Flowchart Sistem Secara Keseluruhan

Flowchart merupakan gambar atau bagan yang menunjukkan urutan dan hubungan antar proses beserta instruksinya. Bagan dinyatakan dengan menggunakan simbol. Setiap simbol memiliki makna proses tertentu. Menghubungkan setiap simbol dengan menggunakan garis penghubung. *Flowchart* merupakan langkah awal dari pembuatan program.

Pengolahan dengan komputer, dapat dirangkum urutan dasar untuk pemecahan suatu masalah yaitu :





- a. **START** : instruksi untuk persiapan peralatan yang diperlukan sebelum menangani pemecahan masalah.
- b. **READ** : instruksi untuk membaca data dari suatu peralatan input.
- c. **PROCESS** : kegiatan yang berkaitan dengan pemecahan persoalan sesuai dengan data yang dibaca.
- d. **WRITE** : instruksi untuk merekam hasil kegiatan ke peralatan *output*.
- e. **END** : mengakhiri kegiatan pengolahan.

Dalam pembuatan program untuk tugas akhir ini menggunakan *Software CodeVision AVR* versi 2.05.3



Gambar 1.10 Tampilan Awal CVAVR

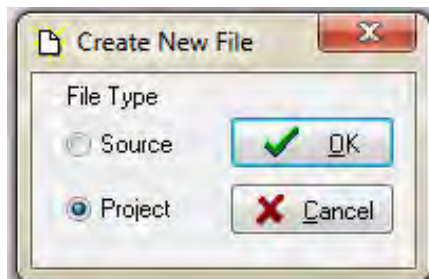
Dari Gambar 3.10 dapat dilihat tampilan awal *software* codevision AVR masih belum terisi program. Sebelum menjelaskan tentang bagaimana cara memprogram menggunakan *software* ini, maka terlebih dahulu dijelaskan beberapa simbol yang sering digunakan pada *software* ini yaitu :

1.  Merupakan simbol *create new project*. Simbol ini berfungsi untuk memulai sebuah *project* program baru.
2.  Simbol disamping adalah simbol *open file atau project*. Simbol ini berfungsi untuk membuka file atau *project* yang telah dibuat sebelumnya.
3.  Simbol ini merupakan simbol *save file*. Simbol ini berfungsi untuk menyimpan file atau *project* yang sedang dibuat.
4.  Merupakan simbol *build all projects file*. Simbol ini berfungsi untuk build program, apakah program yang dibuat telah benar atau masih terdapat error.

1.4.2 Perancangan Program Kontrol Kecepatan Motor DC

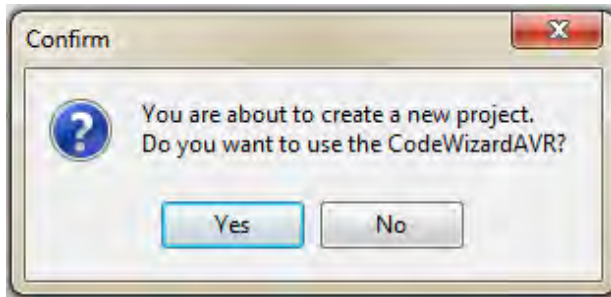
Pada tahapan ini menggunakan *library* yang sudah tersedia pada *software* Codevision AVR. Berikut ini adalah langkah – langkah membuat program kontrol laju kecepatan motor dc :

- a. Klik simbol *create new project* atau seperti pada penjelasan sebelumnya. Setelah klik simbol *create new project*, maka akan muncul *pop up* seperti pada Gambar 3.11 :



Gambar 1.11 Tampilan Membuka *Project* Baru CVAVR

- b. Setelah itu klik ok, kemudian klik yes pada dialog box yang akan muncul seperti pada Gambar 3.12 :



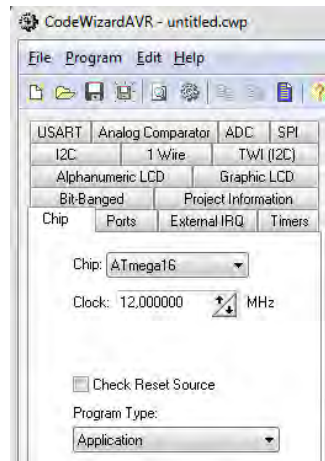
Gambar 1.12 Tampilan Dialog Box Code Wizard CVAVR

- c. Selanjutnya akan muncul dialog box seperti pada Gambar 3.13. Pada dialog tersebut pilih AVR chip type AT90, ATtiny, ATmega, FPSLIC. Kemudian klik ok. Pada tugas akhir ini menggunakan *type* mikrokontroler ATmega 16.



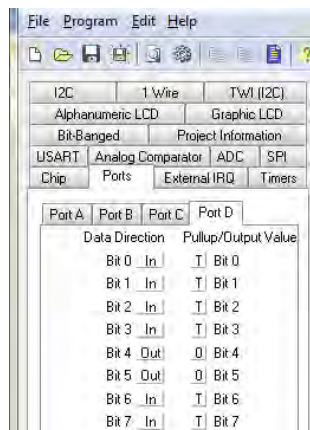
Gambar 1.13 Tampilan Type *Chip* Mikrokontroler

- d. Selanjutnya akan muncul code wizard AVR. Pada tahap ini tersedia beberapa menu *library* yang akan digunakan untuk membuat program. Sebelum membuat program, diperlukan untuk mengatur konfigurasi ATmega yang akan digunakan seperti *chip*, kecepatan *clock*, konfigurasi *port*. Pada tugas akhir ini konfigurasi ATmega 16 dengan kecepatan *clock* sebesar 12 MHz, seperti Gambar 3.14 :



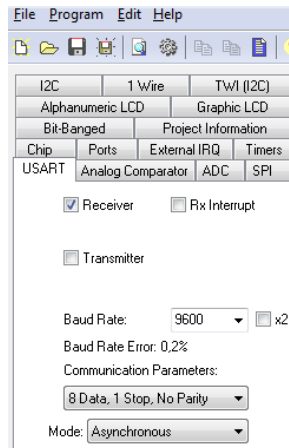
Gambar 1.14 Pengaturan Mikrokontroler dan *Clock*

- e. Setelah melakukan konfigurasi pada *library chip*, selanjutnya melakukan konfigurasi pada *port* ATmega 16. *Port* yang digunakan sebagai *output* yaitu *port* D. Tampilan konfigurasi dari program seperti pada Gambar 3.15.



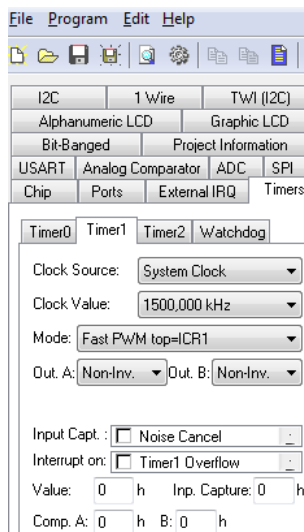
Gambar 1.15 Tampilan Konfigurasi *Port* D ATmega 16

- f. Kemudian pengaturan komunikasi serial *receiver* dan *transmitter*. *Receiver* diaktifkan sedangkan *transmitter* dalam keadaan *off*. Tampilan seperti pada Gambar 3.16.



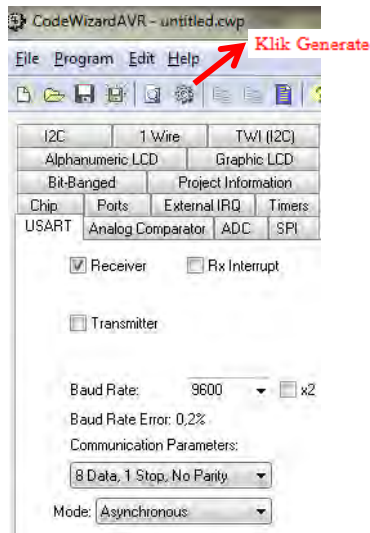
Gambar 1.16 Tampilan Pengaturan Komunikasi Serial

7. Pengaturan pada *library timers*, pengaturan hanya pada *timer 1*. Untuk *timer 0* dan *2* tidak perlu di atur karena hanya *timer 1* yang digunakan. Berikut ini tampilan dari pengaturan *library* seperti pada Gambar 3.17.



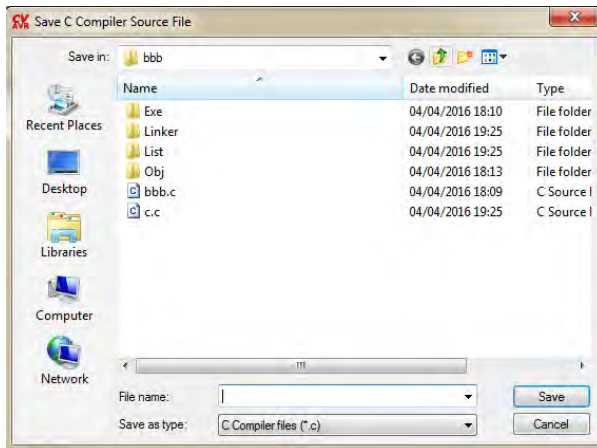
Gambar 1.17 Tampilan Pengaturan Pada *Library Timers*

- h. Setelah semua konfigurasi telah selesai dilakukan, klik simbol *generate* seperti pada Gambar 3.18 :



Gambar 1.18 Tampilan Codewizard AVR

- i. Langkah selanjutnya yaitu menyimpan *file project* tersebut. Penyimpanan *file* terbagi menjadi 3 yaitu penyimpanan dengan ekstensi (.C), (.prj), dan (.cwp). Dalam memberikan nama pada ketiga jenis ekstensi *file* tersebut sebaiknya menggunakan nama yang sama. Penggunaan nama *file* yang sama akan mempermudah dalam pencarian *file*. Gambar 3.19 menunjukkan tampilan *pop up* untuk menyimpan *file*.



Gambar 1.19 Tampilan Menyimpan File C Compiler

- j. Setelah proses penyimpanan *file* selesai, maka akan muncul tampilan awal program seperti Gambar 3.20 :

```

Notes  c.c  aaa.c  bismillah.c
1
2  .....
3  s program was produced by the
4  eWizardAVR V2.05.3 Standard
5  omatic Program Generator
6  opyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.
7  p://www.hpinfotech.com
8
9  ject :
10 sion :
11 e : 4/23/2016
12 hor : iqom
13 pany : twinsRobo
14 ments:
15
16 p type : ATmega16
17 ram type : Application
18 Core Clock frequency: 12.000000 MHz
19 ory model : Small
20 xternal RAM size : 0
21 e Stack size : 256
22 ...../
23

```

Gambar 1.20 Tampilan Program Awal

- k. Untuk *listing* program secara keseluruhan terdapat pada lembar lampiran bagian *listing* program.

1.4.3 Perancangan Program Aplikasi *Smartphone*

Pembuatan *user interface* atau tampilan *smartphone* yaitu desain aplikasi yang nampak pada *smartphone*. Aplikasi ini merupakan media pengguna untuk melihat dan berinteraksi dengan aplikasi.

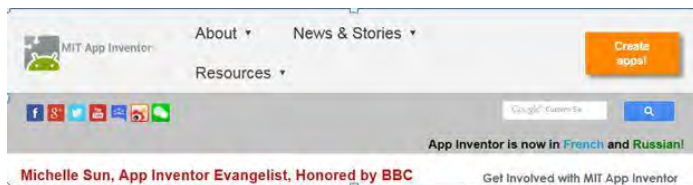
Pemrograman untuk aplikasi *smartphone* dibuat pada aplikasi MIT App Inventor.

MIT App Inventor memiliki empat region yaitu :

1. *Palette* : bagian yang menyediakan komponen yang ingin digunakan untuk membangun *user interface*. *Palette* ini dibagi menjadi beberapa bagian sesuai dengan jenis komponen.
2. *Viewer* : bagian yang merepresentasikan layar *smartphone* dan digunakan untuk tata letak komponen *User interface*.
3. *Components pane* : bagian yang memperlihatkan pandangan berbeda yang ditempatkan di *Viewer*. Pengguna dapat mengatur pandangan agar tumpukan komponen tersusun secara teratur.
4. *Properties pane* : bagian yang berisi daftar semua pengaturan untuk komponen yang saat itu kita pilih.

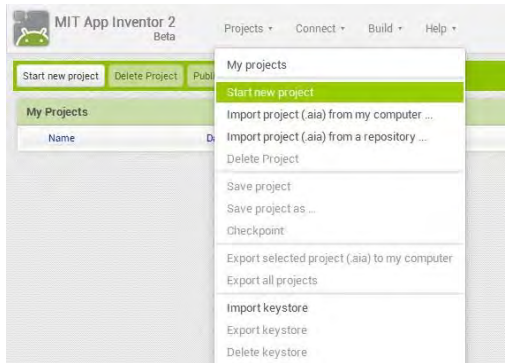
Tahapan untuk membuat *user interface* pada aplikasi *smartphone* yaitu sebagai berikut :

1. Langkah pertama yaitu membuka website dari MIT App Inventor yaitu appinventor.mit.edu. Pembuatan aplikasi pada MIT App Inventor dapat dilakukan secara *online* maupun *offline*. Namun, pada penggunaan secara *offline* akan membutuhkan proses yang panjang. Klik *create app* yang terletak pada bagian kanan atas tampilan website. Tampilan awal dari website ditunjukkan pada Gambar 3.21.



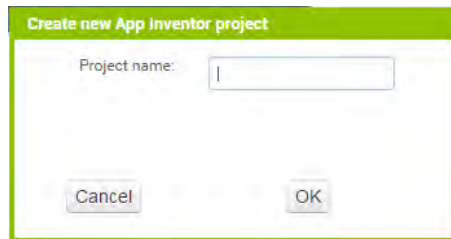
Gambar 1.21 Tampilan Website MIT App Inventor

2. Setelah itu akan muncul halaman utama dari MIT App Inventor. Untuk membuat aplikasi baru klik *Project* lalu klik *start new project*. Tampilan ditunjukkan pada Gambar 3.22.



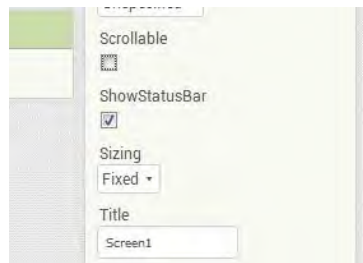
Gambar 1.22 Tampilan Awal MIT App Inventor

3. Kemudian akan muncul tampilan dari *new project* seperti pada Gambar 3.23. Masukkan nama project yang akan dibuat.



Gambar 1.23 Tampilan *Project Name*

4. Lakukan konfigurasi pada bagian layar *smartphone*. *Setting* layar *smartphone* agar tidak dapat discroll. *Setting* ditunjukkan pada Gambar 3.24.



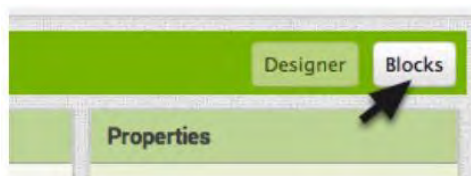
Gambar 1.24 Tampilan *Setting*

5. Tambahkan komponen canvas dengan cara klik *pallette* kemudian klik sub menu *drawing and animation*. Drag and drop komponen canvas ke layar *smartphone*. Untuk menambahkan komponen *accelerometer* klik menu *pallette* lalu klik sub menu sensor. Drag and drop *accelerometer* ke layar *smartphone*. Pada menu *pallette* terdapat beberapa sub menu, sub menu tersebut dapat dilihat pada Gambar 3.25.



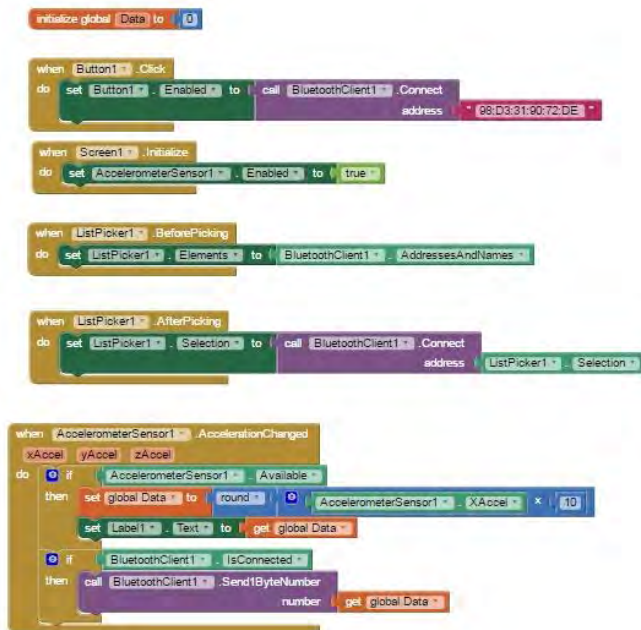
Gambar 1.25 Tampilan Sub Menu Sensor

6. Atur panjang dan lebar canvas agar menyesuaikan pada layar *smartphone* pengguna.
7. Setelah sudah mendesain tampilan utama aplikasi anda, sekarang atur Blocks Editor nya untuk memprogram aplikasi tersebut. Klik pada blocks seperti pada Gambar 3.26.



Gambar 1.26 Tampilan Block Editor

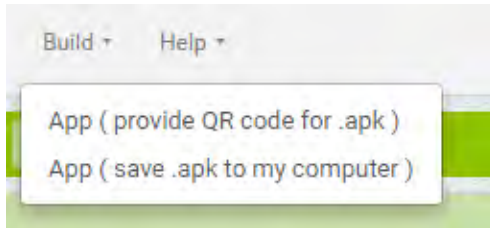
8. Langkah selanjutnya membuat *coding* program untuk sensor *accelerometer*. *Coding* program secara lengkapnya ditunjukkan pada Gambar 3.27.



Gambar 1.27 Tampilan Coding Program

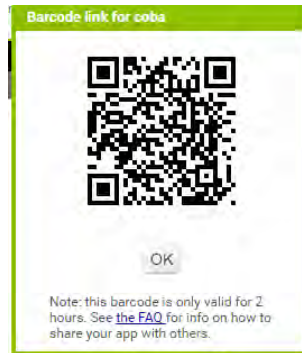
Pada Gambar 3.27 menunjukkan *listing* program untuk mendeteksi *accelerometer*. Pada baris kedua, *listing* untuk mengkoneksikan *bluetooth* dengan alamat *bluetooth* yaitu 98:D3:31:90:72:DE. Jika telah terkoneksi dengan benar maka akan mendeteksi sensor *accelerometer*. Baris ke tiga, menunjukkan program pencarian nama *bluetooth* yang dituju diantara alamat *bluetooth* yang tersedia. Jadi tidak menunjuk pada satu alamat *bluetooth* saja. Baris terakhir adalah penggalan program yang berfungsi merekam data dari sensor *accelerometer*. Nilai sensor *accelerometer* tidak stabil dan mudah berubah, untuk mengatasinya maka diambil 10 data dari sensor. Pembacaan data data sensor dibatasi maksimal hingga 100 data sensor *accelerometer*. Data akan ditampilkan pada *smartphone*.

9. Setelah selesai membuat program, Simpan file dari aplikasi ke komputer. Dengan cara klik *build* pilih app (save.apk to my computer) yang ditunjukkan seperti pada Gambar 3.28.



Gambar 1.28 Tampilan Menu *Build*

Jika ingin menyimpan aplikasi pada *smartphone* dapat dilakukan dengan cara klik menu build sprovide QR code for .apk. Maka akan muncul *barcode* dari program yang telah dibuat seperti pada Gambar 3.29.



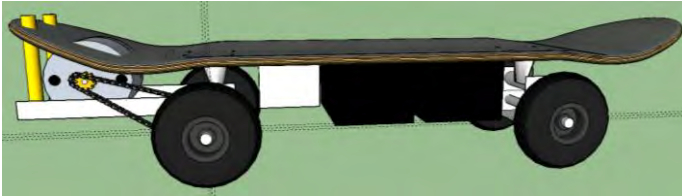
Gambar 1.29 Tampilan *Barcode Link*

1.5 Perancangan *Hardware Skateboard Elektrik*

Pada perancangan mekanik tugas akhir ini adalah *skateboard* elektrik. *Skateboard* elektrik ini berfungsi untuk melekatkan motor dc beserta perangkat elektronik dari tugas akhir ini. Pada perancangan *hardware* dibutuhkan papan *skateboard*, roda, *deck*, dan *truck*.

Papan *skateboard* atau biasa dikenal dengan istilah *deck* terbuat dari kayu maple yang biasanya terdiri dari 7 sampai 9 lapisan kayu yang direkatkan dengan lem dan di press kedalam wadah press untuk membentuk concaves (lekukan pada deck). Papan skate memiliki ukuran yaitu panjang 27 inch. Dibawah papan akan dipasang 4 roda dengan diameter 5 inch atau sekitar 12,7 cm.

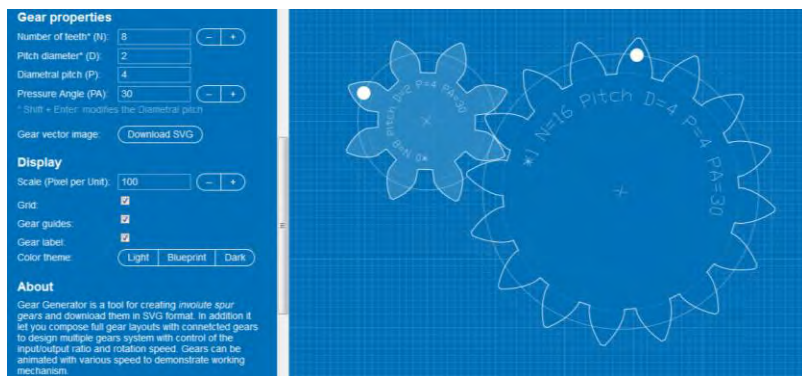
Tiap ujung *deck* dipasang 2 roda, untuk menaruh roda dan untuk menyangga papan *skateboard* akan dipasang *truck* pada kedua ujung papan. Salah satu ujung papan *skateboard* dilubangi untuk melekatkan motor dc. Dibutuhkan ring dan baut untuk mengaitkan antara komponen satu dengan lain. Rangkaian elektronik serta *supply* dilekatkan dibawah papan *skateboard*. Desain *skateboard* dapat dilihat pada Gambar 3.30.



Gambar 1.30 Desain *Skateboard* Elektrik

Pemilihan ukuran roda disesuaikan dengan gear. Roda dan gear akan saling berhubungan agar kecepatan sesuai dengan yang diharapkan. Berikut ini merupakan penjelasan mengenai perhitungan antara kecepatan motor dengan ukuran roda serta rasio gear yang digunakan. Rancangan perbandingan ukuran gear dengan roda ditunjukkan pada Gambar 3.31. Pemilihan roda dilakukan dengan membandingkan hasil kecepatan antara ukuran roda dan gear yang akan digunakan. Perhitungan dijelaskan seperti pada Tabel 3.4.

$$\begin{aligned}
 V &= \frac{\text{keliling roda} \times \text{RPM} \times 60}{10^5} \\
 &= \frac{\pi \times D \times \text{RPM} \times 60}{10^5} \\
 &= \frac{3,14 \times 15,24 \times 2750 \times 60}{10^5} \\
 &= \frac{7.895.844}{10^5} \\
 &= 78,9 V
 \end{aligned}$$



Gambar 1.31 Perancangan Gear pada Roda

Tabel 1.4 Tabel Kecepatan Sesuai dengan Ukuran Diameter Roda

Rasio Gear	Kecepatan (km/jam)			
	3"(7,62cm)	4"(10,16 cm)	5"(12,7 cm)	6" (15,24 cm)
1:1	39,4	52,6	65,7	78,9
1:2	19,7	26,3	32,8	39,4
1:3	9,8	13,1	16,4	19,7
1:4	4,9	6,5	8,2	9,8
1:5	2,4	3,2	4,1	4,9

Dari Tabel 3.4 menunjukkan ukuran roda antara 3,4,5,dan 6 inch terhadap rasio gear yang digunakan antara 1 hingga 5 yang akan mempengaruhi kecepatan motor. Ukuran roda yang dipilih yaitu 5 inch dengan rasio gear 1 : 2 sehingga kecepatan motor maksimal yang diharapkan yaitu 32,8 km/jam.

-----Halaman ini sengaja dikosongkan-----

BAB IV

PENGUJIAN DAN ANALISA DATA

Dalam perencanaan dan pembuatan suatu sistem diperlukan uji coba terhadap alat. Uji coba dilakukan perangkat keras serta perangkat lunak alat. Uji coba dilakukan untuk mengetahui kesesuaian antara teori dengan hasil perancangan, yaitu dengan mengetahui hasil pengukuran pada setiap perangkat yang telah dibuat. Pada bab ini membahas tentang pengujian serta pengukuran dari perangkat keras dan perangkat lunak pada alat serta sistem secara keseluruhan.

2.1 Hasil Perancangan Alat



Gambar 2.1 Tampilan Pada Bagian Samping *Skateboard*

Pada Gambar 4.1 menunjukkan hasil dari perancangan *skateboard* elektrik. *Skateboard* dapat dioperasikan dengan menggunakan *smartphone* yang sudah terdapat aplikasi. Aplikasi ini akan membaca data dari sensor *accelerometer* yang terdapat pada *smartphone*. Cara kerja dari alat ini yaitu ketika *smartphone* digerakkan maka data sensor *accelerometer* akan terbaca oleh aplikasi. Data tersebut akan dikirim pada mikrokontroler dan akan dijadikan sebagai *input driver* motor. *Driver* motor berfungsi untuk mengatur kecepatan motor dengan teknik PWM. Semakin besar data *accelerometer* yang terbaca maka semakin besar pula kecepatan motor. Pada *skateboard* hanya terdapat satu motor

dc yang digunakan untuk menggerakkan 4 roda *skateboard*. *Skateboard* hanya dapat bergerak maju kedepan, jika pada jalan yang berbelok maka dapat dilakukan secara manual.



Gambar 2.2 Tampilan Pada Bagian Atas *Skateboard*



Gambar 2.3 Tampilan *Skateboard* Secara Keseluruhan



Gambar 2.4 Tampilan Pada Bagian Bawah *Skateboard*

Pada Gambar 4.4 menunjukkan bagian bawah dari *skateboard*. Terdapat 4 buah roda masing – masing berukuran 5 inch. Pada salah satu ujung *skateboard* terdapat motor dc 250 watt. Diletakkan rangkaian elektronik *skateboard* serta baterai pada ruang kosong yang berada diantara roda *skateboard*. Peletakkan rangkaian elektronik di bawah *deck* ini bermaksud untuk tidak mengganggu pengguna saat mengendarai *skateboard*. Rangkaian elektronik tersebut diletakkan pada kotak akrilik yang disesuaikan dengan ukuran rangkaian elektronik. Hal ini dimaksudkan untuk melindungi rangkaian elektronik dari kerusakan akibat terkena genangan air yang ada di jalanan.

2.2 Pengujian Motor DC

Pada awal pengujian digunakan motor DC *geared*. Motor dc ini digunakan untuk menggerakkan roda pada *skateboard*. Motor dc ini memenuhi spesifikasi namun motor ini termasuk jenis motor starter. Motor dc starter merupakan motor yang digunakan untuk memberikan putaran awal bagi engine agar dapat menjalankan siklus kerjanya. Motor starter ini biasa digunakan untuk pemicu pada sepeda motor.

Motor dc tidak tahan lama karena motor dc ini hanya dapat digunakan ± 11 menit. Waktu tersebut dirasa kurang memenuhi spesifikasi dari alat ini. *Skateboard* diharapkan dapat dioperasikan dalam waktu yang lama dikarenakan *skateboard* ini akan digunakan sebagai alat transportasi jarak dekat sehingga dibutuhkan motor yang

tahan lama agar dapat menggerakkan skateboard dalam beberapa kali penggunaan setidaknya untuk perjalanan pergi dan pulang. Kekurangan pada motor dc yang tidak dapat digunakan terus menerus pada jangka waktu yang lama inilah yang menjadi alasan untuk mengganti motor dc tersebut.

Motor dc MY1025 dapat digunakan pada jangka waktu yang lama yaitu ± 21 menit. Sehingga lebih efisien jika dibandingkan dengan motor dc starter. Selain itu kecepatan dari motor ini lebih besar jika dibandingkan dengan motor starter yaitu sekitar 2750 RPM sedangkan motor starter hanya 500 RPM. Meskipun kedua motor tersebut memiliki torsi maksimal yang sama yaitu 100 kg.cm.

2.3 Pengujian *Regulator*

Pertama kali pengujian alat menggunakan *regulator* 7805. *Regulator* berfungsi untuk menstabilkan tegangan dari catu daya bila terjadi perubahan tegangan. *Regulator* 7805 ini dapat menghasilkan tegangan output stabil 5 volt. *Regulator* ini dibutuhkan untuk mengoperasikan mikrokontroler. Pada percobaan yang dilakukan pada *regulator* ini adalah *regulator* 7805 tidak dapat bekerja dengan baik sehingga mengakibatkan kerusakan IC pada *driver* motor. Kerusakan ini terjadi dikarenakan arus yang mengalir pada sumber lebih dari 1A menyebabkan IC pada *regulator* tidak dapat menahan arus sehingga mengakibatkan kerusakan. IC 7805 ini sering digunakan pada mikrokontroler, namun batas arus yang terlalu kecil inilah yang membuat *regulator* perlu diberikan tambahan *regulator* agar dapat bekerja pada arus yang besar. Maka digunakanlah *regulator* LM2596. Pada *regulator* LM2596 ini memiliki batas arus maksimal yaitu 43 A. Sedangkan arus yang mengalir pada motor dc adalah 14 A.

2.4 Pengujian *Output* Catu Daya

Terdapat tiga tegangan keluaran pada baterai yang digunakan untuk catu daya pada peralatan. Pengujian ini menggunakan multimeter digital “SANWA” yang digunakan sebagai alat ukur. Probe berwarna merah dihubungkan pada terminal tegangan *output* dan probe berwarna hitam dihubungkan dengan terminal *ground*. Hasil pengujian tegangan *output* pada *power supply* ini terdapat pada Tabel 4.1.

Tabel 2.1 Hasil Pengukuran Tegangan *Output* Baterai (*accu*)

No	Vout Perancangan	Vout Terukur	Error (%)	Keterangan
1	5 Volt	4,98 Volt	0,16%	Sebagai <i>supply</i> tegangan pada mikeokontroller
2	12 Volt	11,16 Volt	3,3%	Sebagai catu daya pada <i>regulator</i>
3	24 Volt	23,3 Volt	2,9%	Sebagai catu daya motor dc

Dari data pengujian tegangan *output* baterai di atas, % error didapatkan dengan menggunakan rumus pada persamaan 4.1 sebagai berikut. Error pada pengukuran disebabkan karena komponen *regulator* yang tidak akurat.

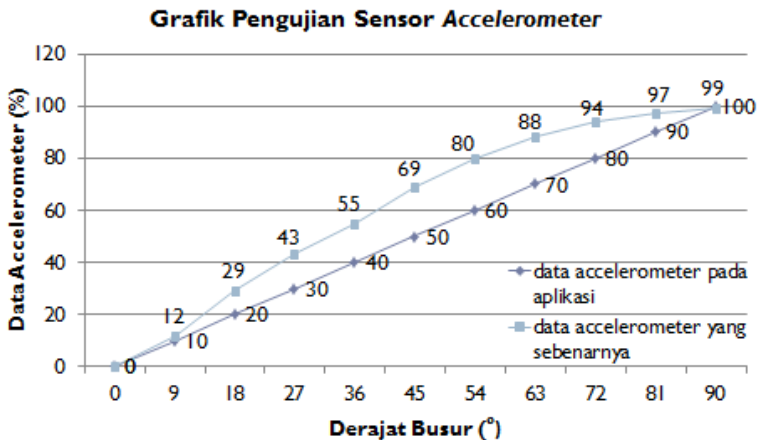
$$\%error = \frac{V \text{ out Terukur} - V \text{ out perancangan}}{V \text{ out perancangan}} \times 100\% \dots\dots\dots(4.1)$$

2.5 Pengujian Sensor *Accelerometer*

Pengujian sensor *accelerometer* yang terdapat pada smartphone dilakukan untuk mengetahui perbandingan antara besar sudut yang terbaca oleh aplikasi dengan hasil yang diharapkan. Pengujian dilakukan dengan membandingkan hasil keduanya dengan menggunakan alat bantu busur derajat. Data sensor *accelerometer* dibuat dalam satuan persen karena sensor *accelerometer* sebenarnya bekerja dengan membaca percepatan alat ketika terjadi perpindahan, bukan membaca perubahan posisi (derajat) secara langsung. Oleh karena itu, nilai sensor *accelerometer* ditampilkan dalam bentuk persen dimana 100% nilai *accelerometer* sama dengan 90⁰ perpindahan. Sensor *accelerometer* yang digunakan hanyalah nilai pembacaan dari X *axis*, sedangkan Y *axis* dan Z *axis* tidak digunakan. Pada Tabel 4.2 menunjukkan hasil antara derajat nilai perpindahan menggunakan busur derajat dengan nilai sensor *accelerometer* dalam persen.

Tabel 2.2 Hasil Pengujian Sensor *Accelerometer*

Derajat busur ($^{\circ}$)	Data <i>Accelerometer</i> yang diharapkan (%)	Data <i>accelerometer</i> yang terukur (%)	Error (%)
0	0	0	0
9	10	12	20
18	20	29	45
27	30	43	43,33
36	40	55	37,5
45	50	69	38
54	60	80	33,33
63	70	88	25,71
72	80	94	17,5
81	90	97	7,77
90	100	99	1



Gambar 2.5 Grafik Pengujian Sensor *Accelerometer*

Pada Gambar 4.5 diatas menunjukkan grafik dari hasil sensor *accelerometer* yang terbaca oleh aplikasi dan hasil sensor *accelerometer* yang sebenarnya. Dari grafik tersebut dapat dilihat bahwa data yang terbaca oleh aplikasi dengan data yang sesungguhnya memiliki

perbedaan. Pengambilan data *accelerometer* dengan menggunakan nilai presentase dimana nilai presentase tertinggi yaitu 100% yang bernilai 90° . Jika dihitung nilai error sensor yang digunakan, maka didapatkan error maksimal pada busur 18° yaitu sebesar 45%.



Gambar 2.6 Pengujian Data Sensor *Accelerometer*

Pada Gambar 4.6 menunjukkan cara pengambilan data sensor *accelerometer*. Pengambilan data dilakukan dengan menentukan posisi dari sesnsor *accelerometer* dari *smartphone* sesuai yang diinginkan setelah itu pertahankan posisi dari *smartphone* tersebut. Kemudian bandingkan posisi yang terbaca pada *smartphone* dengan data sensor *accelerometer* yang terukur dengan data *accelerometer* yang diharapkan.

2.6 Pengujian RPM

Pengujian terhadap banyaknya jumlah putar permenit dilakukan dengan menggunakan alat pengukur RPM yaitu tachometer. Tachometer diarahkan pada poros salah satu roda saat menghitung RPM. Tachometer akan mengeluarkan sinar laser saat dioperasikan. Perhitungan RPM dengan mengambil 10 data *accelerometer* pertama selanjutnya 10 data kedua dst. Hasil pengujian perhitungan RPM ditunjuk kan pada Tabel 4.3.

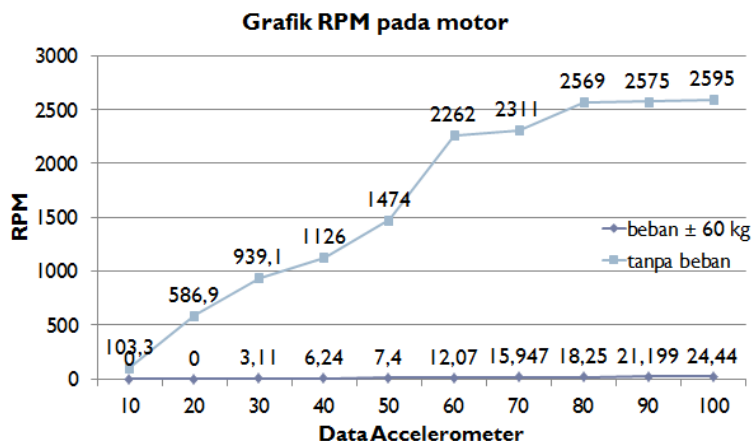
Tabel 2.3 Hasil Pengukuran RPM Tanpa Beban

No.	Data <i>Accelerometer</i> (%)	RPM
1.	10	103,3
2.	20	586,9
3.	30	939,1
4.	40	1126
5.	50	1474
6.	60	2262
7.	70	2311
8.	80	2569
9.	90	2575
10.	100	2595

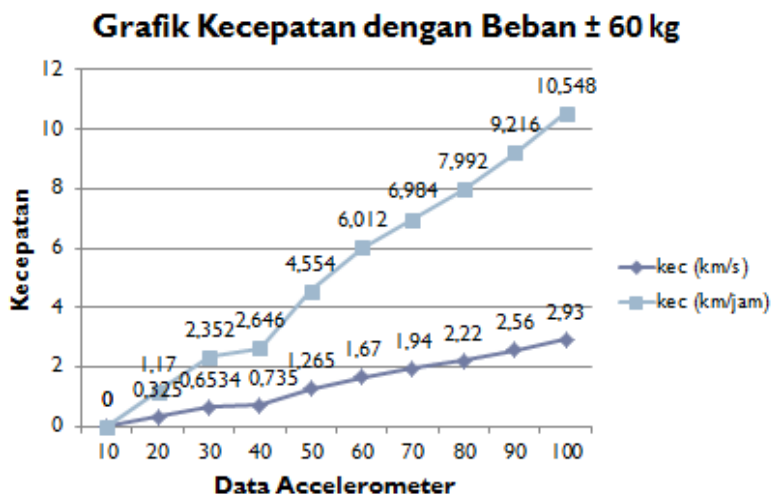
Pada Tabel 4.4 menunjukkan data hasil pengambilan data RPM dengan beban ± 60 kg. Dari Tabel tersebut didapatkan hasil kecepatan maksimal dari motor yaitu 10,5 km/jam. Data RPM diambil dengan menggunakan tachometer.

Tabel 2.4 Hasil Pengukuran RPM dan Kecepatan dengan Beban

Data <i>accelerometer</i> (%)	RPM	Kecepatan (m/s)	Kecepatan (km/jam)
0	0	0	0
10	0	0	0
20	3,11	0,325	1,17
30	6,24	0,6534	2,352
40	7,4	0,735	2,646
50	12,07	1,265	4,554
60	15,947	1,67	6,012
70	18,25	1,94	6,984
80	21,199	2,22	7,992
90	24,44	2,56	9,216
100	27,89	2,93	10,548



Gambar 2.7 Grafik Perbandingan Nilai RPM dengan Beban \pm 60kg dan Tanpa Beban



Gambar 2.8 Grafik Nilai Kecepatan dengan Beban \pm 60kg

Pada Gambar 4.7 menunjukkan perbandingan nilai RPM motor pada saat kondisi tanpa beban dan kondisi beban \pm 60 kg. Pada data

accelerometer 100% nilai dengan beban sebesar 24,44 sedangkan tanpa beban bernilai 2595. Sedangkan pada Gambar 4.8 menunjukkan kecepatan pada saat beban ± 60 kg. Kecepatan maksimum motor dengan menggunakan beban yaitu $\pm 10,5$ km/jam. Dari grafik tersebut dapat dilihat perbedaan nilai RPM yang cukup jauh. Perbedaan tersebut dipicu oleh beban pada *skateboard*.

2.7 Pengujian Output Tegangan Mikrokontroler

Pengujian ini dilakukan untuk mengetahui besarnya tegangan *output* yang ada pada mikrokontroler. Pengujian ini menggunakan multimeter digital "SANWA". Pengukuran dilakukan hanya pada *Port D*. Probe merah dihubungkan dengan *port D* dan probe hitam dihubungkan dengan pin *ground*. Hasil pengujian tegangan *output* terdapat pada Tabel 4.5.

Tabel 2.5 Hasil Pengukuran Tegangan Output Port D

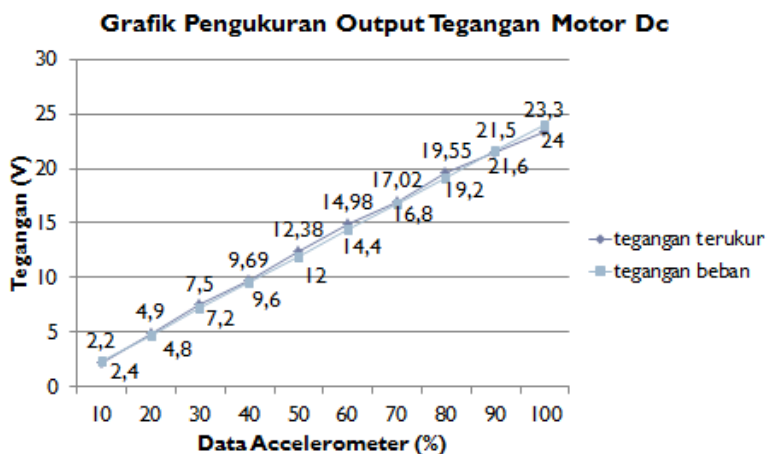
No.	Data Accelerometer(%)	Port D (V)
1.	10	0,07
2.	20	0,23
3.	30	0,38
4.	40	0,31
5.	50	0,38
6.	60	1,50
7.	70	1,61
8.	80	1,71
9.	90	1,78
10.	100	1,81

2.8 Pengujian Output Tegangan Motor DC

Pengujian ini dilakukan untuk mengetahui besarnya tegangan *output* yang ada pada motor dc. Pengujian ini menggunakan multimeter digital "SANWA". Probe merah dihubungkan dengan motor DC dan probe hitam dihubungkan dengan pin *ground*. Hasil pengujian tegangan *output* terdapat pada Tabel 4.6. Kemudian data ini dibandingkan dengan tegangan menggunakan rumus.

Tabel 2.6 Hasil Pengukuran Tegangan Masukan Motor DC

No.	Data <i>Accelerometer</i> (%)	Tegangan Terukur (V)	Tegangan Rumus	Error (%)
1.	10	2,20	2,4	0,0833
2.	20	4,9	4,8	0,0208
3.	30	7,5	7,2	0,0416
4.	40	9,69	9,6	0,0093
5.	50	12,38	12	0,0316
6.	60	14,98	14,4	0,0403
7.	70	17,02	16,8	0,0130
8.	80	19,55	19,2	0,0182
9.	90	21,5	21,6	0,0046
10.	100	23,3	24	0,0292



Gambar 2.9 Grafik Pengujian Output Tegangan Motor DC

Gambar 4.9 menunjukkan grafik pengujian output pada tegangan motor dc yang digunakan pada *skateboard*. Terdapat perbedaan nilai

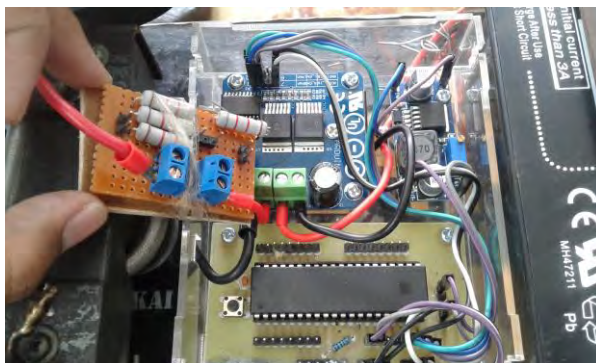
tegangan pada tegangan yang terukur dengan tegangan yang seharusnya. Nilai *error* dari masing – masing pengujian ditunjukkan pada Tabel 4.6, dari tabel tersebut didapatkan rata – rata nilai erornya adalah 0,0058 V.

2.9 Pengujian Arus pada Rangkaian Elektronik



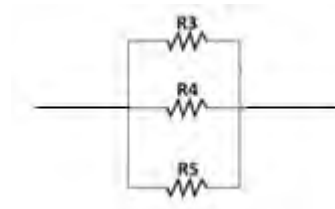
Gambar 2.10. Pengambilan Data Arus Pada Rangkaian Skateboard Elektrik.

Pengujian ini dilakukan untuk mengetahui besarnya arus yang mengalir pada rangkaian. Pengujian ini menggunakan multimeter digital "SANWA", seperti pada Gambar 4.10. Pengukuran arus dilakukan dengan beban kurang lebih 60 kg dan tanpa menggunakan beban. Pengujian dilakukan dengan menghubungkan sumber yang terhubung secara seri dengan Hasil pengujian tegangan arus terdapat pada Tabel 4.7.



Gambar 2.11 Pengambilan Data Arus yang Menggunakan Rangkaian Pembagi Arus.

Pengambilan data arus yang menggunakan rangkaian pembagi arus. Dikarenakan arus yang keluar dari motor DC ± 14 A. Rangkaian pembagi arus ini dilakukan untuk mempermudah pengambilan data arus. Pengambilan data tidak dapat dilakukan dengan menggunakan multimeter secara langsung dikarenakan nilai arus yang mengalir terlalu besar dan melebihi nilai arus pada multimeter. Rangkaian pembagian arus tersebut yaitu :

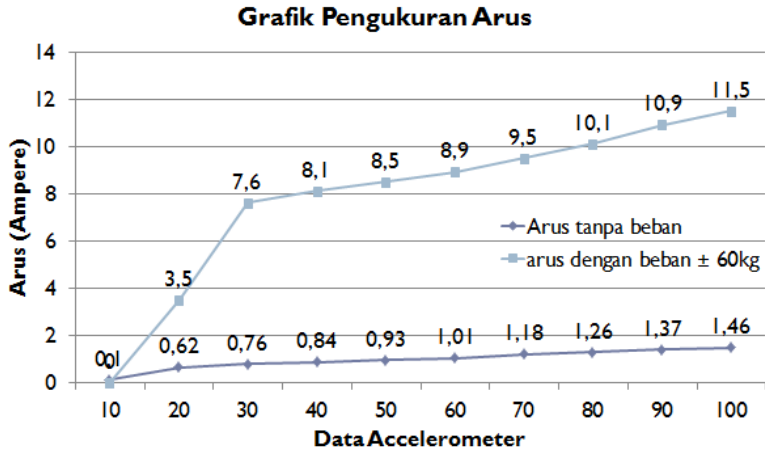


Gambar 2.12 Rangkaian Pembagi Arus

Rangkaian pembagi arus tersebut menggunakan 3 buah resistor dengan nilai $1,2 \text{ k}\Omega$. Arus yang terbaca oleh multimeter hanya untuk satu resistor yang dilewati arus. Arus yang lewat pada ketiga resistor bernilai sama. Pengujian tersebut dapat dilihat pada Gambar 4.11.

Tabel 2.7 Pengujian Arus Tanpa Beban dan dengan Sebesar $\pm 60 \text{ kg}$

No.	Data <i>Accelerometer</i> (%)	Arus tanpa beban (A)	Arus dengan beban (A)
1	0	0,1	0
2.	10	0,62	3,5
3.	20	0,76	7,6
4.	30	0,84	8,1
5.	40	0,93	8,5
6.	50	1,01	8,9
7.	60	1,18	9,5
8.	70	1,26	10,1
9.	80	1,37	10,9
10.	90	1,46	11,5
11.	100	1,58	12,9

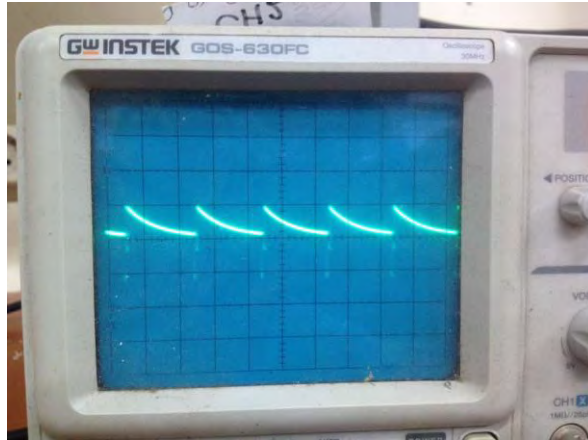


Gambar 2.13 Grafik Pengukuran Arus pada Rangkaian Elektronik

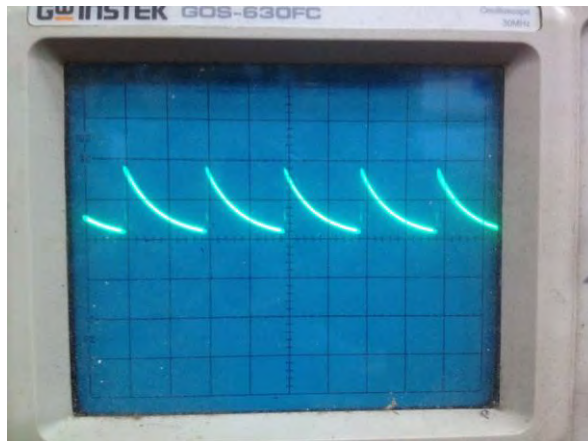
2.10 Pengambilan Data PWM

Pengujian pembangkitan PWM dilakukan untuk mengetahui bagaimana pengaruh penambahan terhadap tegangan yang dihasilkan oleh motor DC. Apabila pemberian tidak sebanding dengan keluaran tegangan, maka dapat menyebabkan motor DC tidak bekerja sesuai dengan yang diinginkan. Pulsa PWM berfungsi mengatur kecepatan motor DC.

Pengambilan bentuk sinyal PWM diperoleh dengan menghubungkan *probe* positif dengan keluaran PWM (*port D4*), dan *probe* negatif dengan *ground*. *Port D4* pada mikrokontroler dapat dimanfaatkan sebagai pembangkit PWM. Osiloskop akan menampilkan respon gelombang PWM yang dibangkitkan. Pengujian pembangkitan PWM ditunjukkan pada Gambar 4.14 dan Gambar 4.15.



Gambar 2.14 Pengambilan Data PWM dengan Hasil Amplitudo 0,5 volt



Gambar 2.15 Pengambilan Data PWM dengan Hasil Amplitudo 1 volt

Pada Gambar 4.14 dan Gambar 4.15 menunjukkan hasil pengaturan PWM dengan pengaturan nilai *accelerometer* dapat dilihat bahwa ketika sensor *accelerometer* bernilai 3 amplitudo sinyal naik hingga satu kotak dan ketika bernilai 6 naik menjadi 2 kotak. Pada osiloskop ini tiap kenaikan satu kotak amplitudo sama dengan kenaikan nilai tegangan 0,5 volt.

2.11 Pengujian Jarak Komunikasi *Bluetooth*

Pada pengujian jarak komunikasi *bluetooth* dilakukan dengan cara menjalankan *skateboard* dengan jarak sejauh mungkin dengan menggunakan *smartphone* sebagai *remote*. Pengujian data jarak jauh agar dapat mengetahui kualitas dari *bluetooth*. Pengujian jarak komunikasi *bluetooth* dilakukan di ruang terbuka tanpa penghalang. Hasil pengujian jarak komunikasi dengan menggunakan *bluetooth* ditunjukkan pada tabel 4.8.

Tabel 2.8 Pengujian Jarak Komunikasi *Bluetooth*

Jarak	Keterangan
0 – 10 meter	Tersambung
11 – 25 meter	Tersambung
21 – 37 meter	Tersambung
> 37 meter	Tidak Tersambung

2.12 Pengujian Aplikasi dengan Menggunakan *Smartphone*

Hasil pengujian pada Tabel 4.9 dilakukan dengan menginstal aplikasi dan menjalankan pada *device* yang berbeda. Pengujian ini untuk memastikan apakah aplikasi maupun *skateboard* dapat dikendalikan dengan semua *device* dengan baik. Hasil pengujian pada Tabel 4.9 menunjukkan tidak semua *device* dapat menjalankan aplikasi dengan baik. Pada *smartphone* oppo dan sony memiliki koneksi data yang kurang cepat. Sedangkan pada xperia sensor *accelerometer* dapat berjalan namun, sensor terlambat untuk menangkap respon. Terdapat jeda waktu dalam setiap perubahan gerakan dari sensor *accelerometer*.

Tabel 2.9 Pengujian Aplikasi Menggunakan *Device* Lain

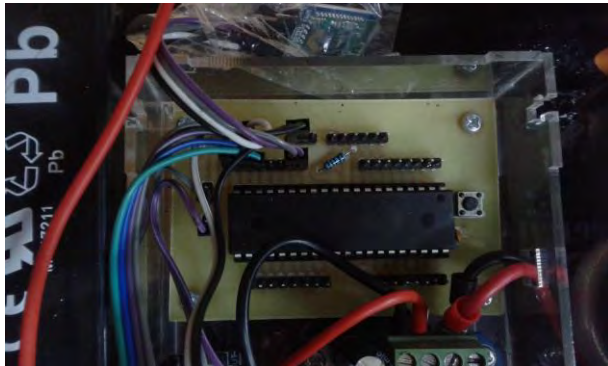
Device	OS	<i>Bluetooth</i>	<i>Accelerometer</i>	Status Pairing	Koneksi Data	Keberhasilan
Samsung Galaxy tab 7.7	4.1	Ok	Ok	Ok	Ok	Ok
Oppo find muse R821	4.2	Ok	Ok	Ok	Lambat	Kurang
Sony Xperia GO ST27i	4.1	Ok	Tidak sesuai	Ok	Lambat	Kurang

LAMPIRAN A

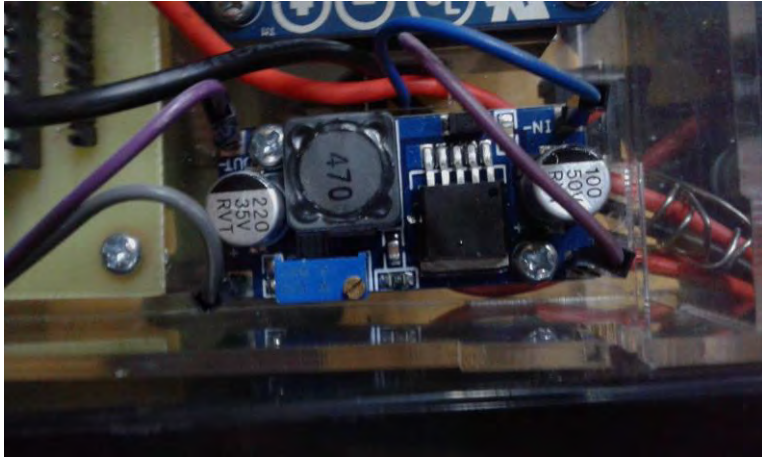
A.1. Tampilan Alat



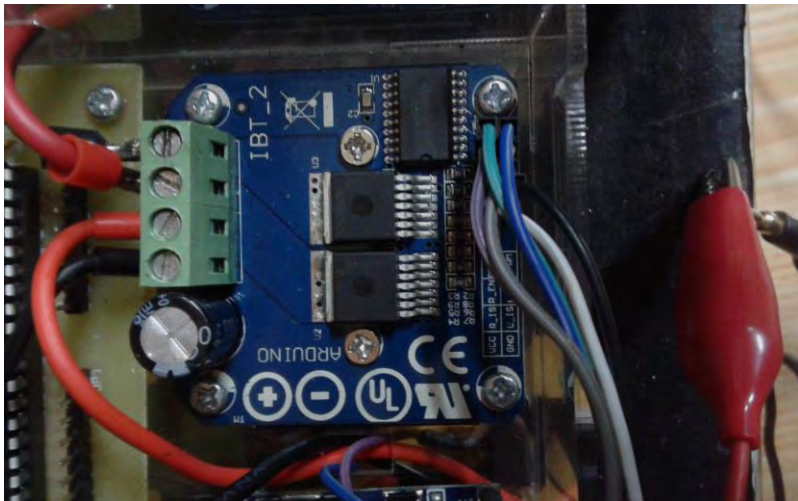
Gambar diatas merupakan rangkaian elektronik dari *skateboard* elektronik. Rangkaian elektronik tersebut diletakkan pada kotak akrilik. Rangkaian elektronik tersebut terdiri dari minimum sistem mikrokontroler ATmega16, dribver motor dc BTS7960, *regulator*, serta modul *bluetooth* HC – 05.



Rangkaian minimum sistem ATmega 16 pada skateboard elektrik. Minimum Sistem mikrokontroler ATmega 16 berfungsi sebagai pusat kendali dari sistem.



Tampilan dari rangkaian *regulator* LM2596. *Regulator* berfungsi sebagai pengatur tegangan yang keluar dari catu daya.



Rangkaian *driver* motor BTS 7960. *Driver* motor digunakan sebagai pengatur kecepatan pada motor DC.



Tampilan dari dua buah *accu* kering dengan nilai *voltage* 12 V, untuk masing – masing *accu* kering. Catu daya digunakan sebagai pemasok tegangan ke seluruh rangkaian elektronik.

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN B

B.1. *Listing* Program dengan CV AVR

```
#include <mega16.h>
#include <delay.h>

#include <stdio.h>

#define enL    PORTD.7
#define enR    PORTD.6
#define pwmka  OCR1B
#define pwmki  OCR1A

void motorMaju(int kec);
void motorMundur();
#ifndef RXB8
#define RXB8 1
#endif

#ifndef TXB8
#define TXB8 0
#endif

#ifndef UPE
#define UPE 2
#endif

#ifndef DOR
#define DOR 3
#endif

#ifndef FE
#define FE 4
#endif

#ifndef UDRE
#define UDRE 5
```

```

#endif

#ifndef RXC
#define RXC 7
#endif

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// USART Receiver buffer
#define RX_BUFFER_SIZE 8
char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE <= 256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
unsigned int rx_wr_index,rx_rd_index,rx_counter;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

int pangkat(int x){
    int i,hasil;
    hasil=1;
    for(i=0;i<x;i++){
        hasil*=10;
    }
    return(hasil);
}
int kiriman=0,array[5],state=0;;
int iterasi=0;
// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
    unsigned char jumlahIterasi,buffer;

```

```

char status,data;
status=UCSRA;
data=UDR;

```

```

        if(data>0&&data<=100)
            kiriman=data;
            state=0;
//      if(data==""){
//          iterasi--;
//          jumlahIterasi=iterasi;
//          buff=0;
//          while(iterasi>=0){
//              buff+=array[jumlahIterasi-iterasi]*pangkat(iterasi);
//              iterasi--;
//          }
//
//          kiriman=buff;
//          putchar(kiriman);
//          iterasi=0;
//      }
//      else if(data>=0x30 && data<0x40){
//          array[iterasi]=data-(0x30);
//          iterasi++;
//      }
//

```

```

}

```

```

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter==0);

```

```

data=rx_buffer[rx_rd_index++];
#if RX_BUFFER_SIZE != 256
if (rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
#endif
asm("cli")
--rx_counter;
asm("sei")
return data;
}
#pragma used-
#endif

// Standard Input/Output functions
#include <stdio.h>

// Declare your global variables here
int i;
float velocity;
void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTB=0x00;
DDRB=0x00;

```

```
// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTC=0x00;
DDRC=0x00;
```

```
// Port D initialization
// Func7=In Func6=In Func5=Out Func4=Out Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=0 State4=0 State3=T State2=T State1=T
State0=T
PORTD=0x00;
DDRD=0xF0;
```

```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 1500.000 kHz
// Mode: Fast PWM top=ICR1
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0xA2;
TCCR1B=0x1A;
```

```

TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x05;
ICR1L=0xdc;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: Off
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;

```

```

UBRR1=0x4D;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Global enable interrupts
#asm("sei")
enL=0;
enR=0;

while (1)
{
    // Place your code here
    if(kiriman>0){
        velocity= (float)(kiriman)*15; // *( 1500/(100-5))
        // velocity=800;
        motorMaju(velocity);
    }

    else {
        motorMundur();
    }
}
//putchar(kiriman);

```



```
    }  
}
```

```
void motorMaju(int kec){  
    enL=1;  
    enR=1;  
    pwmka=kec;  
    pwmki=0;
```

```
}
```

```
void motorMundur(){  
    enL=0;  
    enR=0;  
    pwmki=0;  
    pwmka=0;  
}
```

LAMPIRAN C

C.1. Datasheet Driver Motor DC BTS7960

**High Current PN Half Bridge
NovalithIC™**

BTS 7960B

BTS 7960P

Product Summary

The BTS 7960 is a fully integrated high current half bridge for motor drive applications. It is part of the NovalithIC™ family containing one p-channel highside MOSFET and one n-channel lowside MOSFET with an integrated driver IC in one package. Due to the p-channel highside switch the need for a charge pump is eliminated thus minimizing EMI. Interfacing to a microcontroller is made easy by the integrated driver IC which features logic level inputs, diagnosis with current sense, slew rate adjustment, dead time generation and protection against overtemperature, overvoltage, undervoltage, overcurrent and short circuit.

The BTS 7960 provides a cost optimized solution for protected high current PWM motor drives with very low board space consumption.

**BTS 7960B
P-TO-263-7**



**BTS 7960P
P-TO-220-7**



Basic Features

- Path resistance of typ. 16 mΩ @ 25 °C
- Low quiescent current of typ. 7 μA @ 25 °C
- PWM capability of up to 25 kHz combined with active freewheeling
- Switched mode current limitation for reduced power dissipation in overcurrent
- Current limitation level of 43 A typ.
- Status flag diagnosis with current sense capability
- Overtemperature shut down with latch behaviour
- Overvoltage lock out
- Undervoltage shut down
- Driver circuit with logic level inputs
- Adjustable slew rates for optimized EMI

Type	Ordering Code	Package
BTS 7960B	Q67060-S6160	P-TO-263-7
BTS 7960P	on request	P-TO-220-7

3 Maximum Ratings

-40 °C < T_j < 150 °C (unless otherwise specified)

Pos	Parameter	Symbol	Limits		Unit	Test Condition
			min	max		

Electrical Maximum Ratings

3.0.1	Supply voltage	V_{VS}	-0.3	45	V	
3.0.2	Logic Input Voltage	V_{IN} V_{INH}	-0.3	5.3	V	
3.0.3	HS/LS continuous drain current	$I_{D(HS)}$ $I_{D(LS)}$	-40	40 ¹⁾	A	$T_C < 85^\circ\text{C}$ switch active
3.0.4	HS pulsed drain current	$I_{D(HS)}$	-60	60 ¹⁾	A	$T_C < 85^\circ\text{C}$ $t_{\text{pulse}} = 10\text{ms}$
3.0.5	LS pulsed drain current	$I_{D(LS)}$	-60	60 ¹⁾	A	
3.0.6	Voltage at SR pin	V_{SR}	-0.3	1.0	V	
3.0.7	Voltage between VS and IS pin	$V_{VS} - V_{IS}$	-0.3	45	V	
3.0.8	Voltage at IS pin	V_{IS}	-20	45	V	

Thermal Maximum Ratings

3.0.9	Junction temperature	T_j	-40	150	°C	
3.0.10	Storage temperature	T_{stg}	-55	150	°C	

ESD Susceptibility

3.0.11	ESD susceptibility HBM	V_{ESD}			kV	according to EIA/ JESD 22-A 114B
	IN, INH, SR, IS		-2	2		
	OUT, GND, VS		-6	6		

¹⁾ Maximum reachable current may be smaller depending on current limitation level

Note: Maximum ratings are absolute ratings; exceeding any one of these values may cause irreversible damage to the device. Exposure to maximum rating conditions for extended periods of time may affect device reliability

4 Block Description and Characteristics

4.1 Supply Characteristics

$-40^{\circ}\text{C} < T_j < 150^{\circ}\text{C}$, $8\text{ V} < V_S < 18\text{ V}$, $I_L = 0\text{ A}$ (unless otherwise specified)

-40 °C < T _J < 150 °C, 6 V < V _S < 18 V, I _L = 0 A (unless otherwise specified)							
Pos.	Parameter	Symbol	Limit Values			Unit	Test Conditions
			min.	typ.	max.		
General							
4.1.1	Operating Voltage	V _S	5.5	—	27.5	V	
4.1.2	Supply Current	I _{VS(on)}	—	2	3	mA	V _{INH} = 5 V V _{IN} = 0 V or 5 V R _{SR} =0 Ω DC-mode normal operation (no fault condition)
4.1.3	Quiescent Current	I _{VS(off)}	—	7	15	μA	V _{INH} = 0 V V _{IN} = 0 V or 5 V T _J < 85 °C
			—	—	65	μA	V _{INH} = 0 V V _{IN} = 0 V or 5 V

4.2 Power Stages

The power stages of the BTS 7960 consist of a p-channel vertical DMOS transistor for the high side switch and a n-channel vertical DMOS transistor for the low side switch. All protection and diagnostic functions are located in a separate top chip. Both switches can be operated up to 25 kHz, allowing active freewheeling and thus minimizing power dissipation in the forward operation of the integrated diodes.

The on state resistance R_{ON} is dependent on the supply voltage V_S as well as on the junction temperature T_j . The typical on state resistance characteristics are shown in Figure 4.

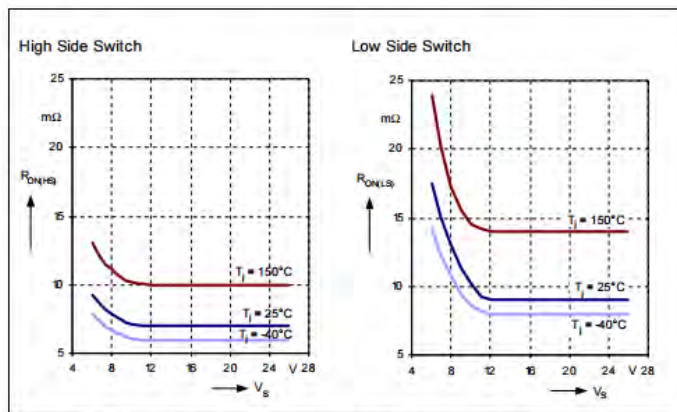


Figure 4 Typical On State Resistance vs. Supply Voltage

4.2.1 Power Stages - Static Characteristics

- 40 °C < T_J < 150 °C, 8 V < V_S < 18 V (unless otherwise specified)

Pos.	Parameter	Symbol	Limit Values			Unit	Test Conditions
			min.	typ.	max.		

High Side Switch - Static Characteristics

4.2.1	On state high side resistance	$R_{ON(HS)}$	-	7	9	mΩ	$I_{OUT} = 9\text{ A}$ $V_S = 13.5\text{ V}$ $T_J = 25\text{ °C}$ $T_J = 150\text{ °C}$
			-	10	12.5		
4.2.2	Leakage current high side	$I_{L(LKHS)}$	-	-	1	μA	$V_{INH} = 0\text{ V}$ $V_{OUT} = 0\text{ V}$ $T_J < 85\text{ °C}$
			-	-	50	μA	$V_{INH} = 0\text{ V}$ $V_{OUT} = 0\text{ V}$ $T_J = 150\text{ °C}$
4.2.3	Reverse diode forward-voltage high side ¹⁾	$V_{DS(HS)}$	-	0.9	1.5	V	$I_{OUT} = -9\text{ A}$ $T_J = -40\text{ °C}$ $T_J = 25\text{ °C}$ $T_J = 150\text{ °C}$
			-	0.8	1.1		
			-	0.6	0.8		

Low Side Switch - Static Characteristics

4.2.4	On state low side resistance	$R_{ON(LS)}$	-	9	12	mΩ	$I_{OUT} = -9\text{ A}$ $V_S = 13.5\text{ V}$ $T_J = 25\text{ °C}$ $T_J = 150\text{ °C}$
			-	14	18		
4.2.5	Leakage current low side	$I_{L(LKLS)}$	-	-	1	μA	$V_{INH} = 0\text{ V}$ $V_{OUT} = V_S$ $T_J < 85\text{ °C}$
			-	-	15	μA	$V_{INH} = 0\text{ V}$ $V_{OUT} = V_S$ $T_J = 150\text{ °C}$
4.2.6	Reverse diode forward-voltage low side ¹⁾	$V_{SD(LS)}$	-	0.9	1.5	V	$I_{OUT} = 9\text{ A}$ $T_J = -40\text{ °C}$ $T_J = 25\text{ °C}$ $T_J = 150\text{ °C}$
			-	0.8	1.1		
			-	0.6	0.8		

¹⁾ Due to active freewheeling, diode is conducting only for a few μs, depending on R_{SR}

4.2.3 Power Stages - Dynamic Characteristics

-40 °C < T_J < 150 °C, $V_S = 13.5$ V, $R_{load} = 2\Omega$ (unless otherwise specified)

Pos.	Parameter	Symbol	Limit Values			Unit	Test Conditions
			min.	typ.	max.		
High Side Switch Dynamic Characteristics							
4.2.7	Rise-time of HS	$t_{r(HS)}$	0.5 — 2.8	1 2 7	1.5 — 11	μs	$R_{SR} = 0 \Omega$ $R_{SR} = 5.1 \text{ k}\Omega$ $R_{SR} = 51 \text{ k}\Omega$
4.2.8	Slew rate HS on	$\Delta V_{OUT}/t_{r(HS)}$	— — —	11 6 1.6	— — —	V/ μs	$R_{SR} = 0 \Omega$ $R_{SR} = 5.1 \text{ k}\Omega$ $R_{SR} = 51 \text{ k}\Omega$
4.2.9	Switch on delay time HS	$t_{dr(HS)}$	1.7 — 5.6	3.1 4.4 14	4.5 — 22.4	μs	$R_{SR} = 0 \Omega$ $R_{SR} = 5.1 \text{ k}\Omega$ $R_{SR} = 51 \text{ k}\Omega$
4.2.10	Fall-time of HS	$t_{f(HS)}$	0.5 — 2.8	1 2 7	1.5 — 11	μs	$R_{SR} = 0 \Omega$ $R_{SR} = 5.1 \text{ k}\Omega$ $R_{SR} = 51 \text{ k}\Omega$
4.2.11	Slew rate HS off	$-\Delta V_{OUT}/t_{f(HS)}$	— — —	11 6 1.6	— — —	V/ μs	$R_{SR} = 0 \Omega$ $R_{SR} = 5.1 \text{ k}\Omega$ $R_{SR} = 51 \text{ k}\Omega$
4.2.12	Switch off delay time HS	$t_{dt(HS)}$	1.2 — 4	2.4 3.4 10	3.6 — 16	μs	$R_{SR} = 0 \Omega$ $R_{SR} = 5.1 \text{ k}\Omega$ $R_{SR} = 51 \text{ k}\Omega$

-40 °C < T_j < 150 °C, $V_S = 13.5$ V, $R_{load} = 2\Omega$ (unless otherwise specified)

Pos.	Parameter	Symbol	Limit Values			Unit	Test Conditions
			min.	typ.	max.		
Low Side Switch Dynamic Characteristics							
4.2.13	Rise-time of LS	$t_{r(LS)}$	0.5 — 2.8	1 2 7	1.5 — 11	μs	$R_{SR} = 0 \Omega$ $R_{SR} = 5.1 \text{ k}\Omega$ $R_{SR} = 51 \text{ k}\Omega$
4.2.14	Slew rate LS switch off	$\Delta V_{OUT}/t_{r(LS)}$	— — —	11 6 1.6	— — —	V/ μs	$R_{SR} = 0 \Omega$ $R_{SR} = 5.1 \text{ k}\Omega$ $R_{SR} = 51 \text{ k}\Omega$
4.2.15	Switch off delay time LS	$t_{dr(LS)}$	0.7 — 2.8	1.3 2.2 7	1.9 — 11.2	μs	$R_{SR} = 0 \Omega$ $R_{SR} = 5.1 \text{ k}\Omega$ $R_{SR} = 51 \text{ k}\Omega$
4.2.16	Fall-time of LS	$t_{f(LS)}$	0.5 — 2.8	1 2 7	1.5 — 11	μs	$R_{SR} = 0 \Omega$ $R_{SR} = 5.1 \text{ k}\Omega$ $R_{SR} = 51 \text{ k}\Omega$
4.2.17	Slew rate LS switch on	$-\Delta V_{OUT}/t_{f(LS)}$	— — —	11 6 1.6	— — —	V/ μs	$R_{SR} = 0 \Omega$ $R_{SR} = 5.1 \text{ k}\Omega$ $R_{SR} = 51 \text{ k}\Omega$
4.2.18	Switch on delay time LS	$t_{df(LS)}$	2.2 — 6.4	4 5.6 16	5.8 — 25.4	μs	$R_{SR} = 0 \Omega$ $R_{SR} = 5.1 \text{ k}\Omega$ $R_{SR} = 51 \text{ k}\Omega$

4.3.6 Electrical Characteristics - Protection Functions

– 40 °C < T_J < 150 °C; 8 V < V_S < 18 V (unless otherwise specified)

Pos.	Parameter	Symbol	Limit Values			Unit	Test Conditions
			min.	typ.	max.		
Under Voltage Shut Down							
4.3.1	Switch-ON voltage	$V_{UV(ON)}$	–	–	5.5	V	V_S increasing
4.3.2	Switch-OFF voltage	$V_{UV(OFF)}$	4.0	–	5.4	V	V_S decreasing
4.3.3	ON/OFF hysteresis	$V_{UV(HY)}$	–	0.2	–	V	–
Over Voltage Lock Out							
4.3.4	Switch-ON voltage	$V_{OV(ON)}$	27.5	–	–	V	V_S decreasing
4.3.5	Switch-OFF voltage	$V_{OV(OFF)}$	27.6	–	30	V	V_S increasing
4.3.6	ON/OFF hysteresis	$V_{OV(HY)}$	–	0.2	–	V	–
Current Limitation							
4.3.7	Current limitation detection level high side	I_{CLH0}	47	62	84	A	$V_S=13.5\text{ V}$
			44	60	80		$T_J = -40\text{ °C}$
			43	59	79		$T_J = 25\text{ °C}$
							$T_J = 150\text{ °C}$
4.3.8	Current limitation detection level low side	I_{CLL0}	36	47	64	A	$V_S=13.5\text{ V}$
			34	43	61		$T_J = -40\text{ °C}$
			33	42	61		$T_J = 25\text{ °C}$
							$T_J = 150\text{ °C}$
Current Limitation Timing							
4.3.9	Shut off time for HS and LS	t_{CLS}	70	115	210	μs	$V_S=13.5\text{ V}$
Thermal Shut Down							
4.3.10	Thermal shut down junction temperature	T_{JSD}	152	175	200	°C	–
4.3.11	Thermal switch on junction temperature	T_{JSO}	150	–	190	°C	–
4.3.12	Thermal hysteresis	ΔT	–	7	–	K	–
4.3.13	Reset pulse at INH pin (INH low)	t_{reset}	3	–	–	μs	–

4.4.5 Truth Table

Device State	Inputs		Outputs			Mode
	INH	IN	HSS	LSS	IS	
Normal operation	0	X	OFF	OFF	0	Stand-by mode
	1	0	OFF	ON	0	LSS active
	1	1	ON	OFF	CS	HSS active
Over-voltage (OV)	X	X	ON	OFF	1	Shut-down of LSS, HSS activated, error detected
Under-voltage (UV)	X	X	OFF	OFF	0	UV lockout
Overtemperature or short circuit of HSS or LSS	0	X	OFF	OFF	0	Stand-by mode, reset of latch
	1	X	OFF	OFF	1	Shut-down with latch, error detected
Current limitation mode	1	1	OFF	ON	1	Switched mode, error detected
	1	0	ON	OFF	1	Switched mode, error detected

Inputs:	Switches	Status Flag IS:
0 = Logic LOW	OFF = switched off	CS = Current sense mode
1 = Logic HIGH	ON = switched on	1 = Logic HIGH (error)
X = 0 or 1		

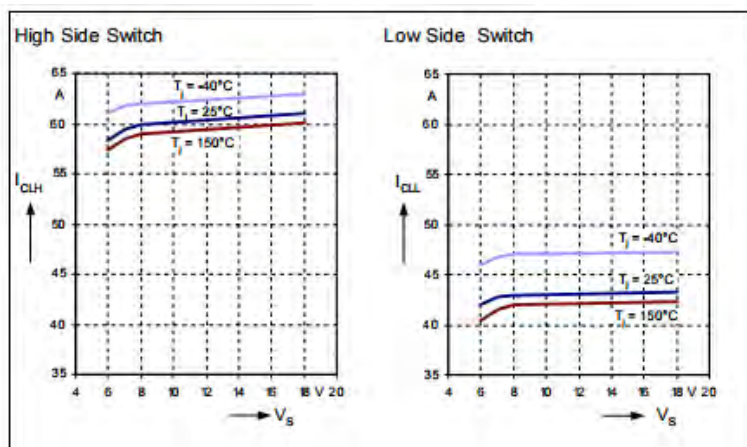


Figure 9 Typical Current Limitation Detection Levels vs. Supply Voltage

In combination with a typical inductive load, such as a motor, this results in a switched mode current limitation. That way of limiting the current has the advantage that the power dissipation in the BTS 7960 is much smaller than by driving the MOSFETs in linear mode. Therefore it is possible to use the current limitation for a short time without exceeding the maximum allowed junction temperature (e.g. for limiting the inrush current during motor start up). However, the regular use of the current limitation is allowed as long as the specified maximum junction temperature is not exceeded. Exceeding this temperature can reduce the lifetime of the device.

4.3.5 Short Circuit Protection

The device is short circuit protected against

- output short circuit to ground
- output short circuit to supply voltage
- short circuit of load

The short circuit protection is realized by the previously described current limitation in combination with the over-temperature shut down of the device

6.1 Application Example

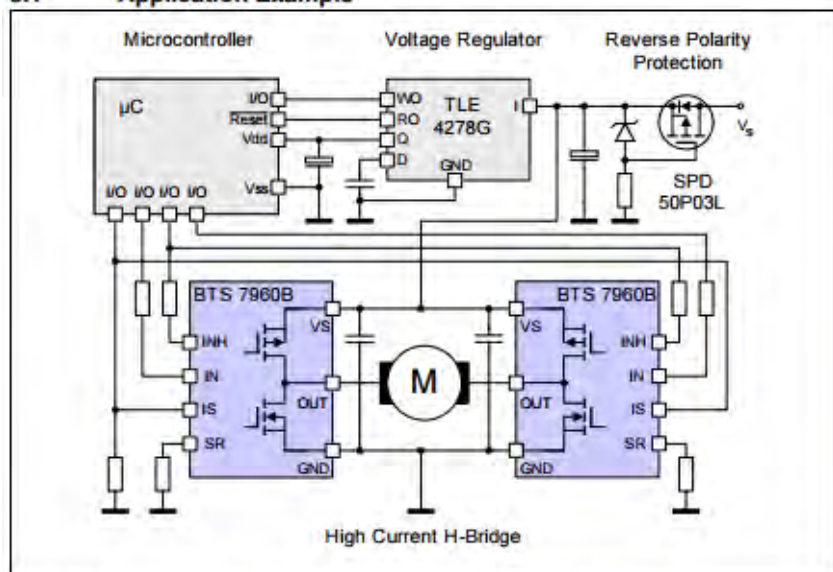


Figure 11 Application Example: H-Bridge with two BTS 7960B

6.2 Layout Considerations

Due to the fast switching times for high currents, special care has to be taken to the PCB layout. Stray inductances have to be minimized in the power bridge design as it is necessary in all switched high power bridges. The BTS 7960 has no separate pin for power ground and logic ground. Therefore it is recommended to assure that the offset between the ground connection of the slew rate resistor, the current sense resistor and ground pin of the device (GND / pin 1) is minimized. If the BTS 7960 is used in a H-bridge or B6 bridge design, the voltage offset between the GND pins of the different devices should be small as well.

A ceramic capacitor from V_s to GND close to each device is recommended to provide current for the switching phase via a low inductance path and therefore reducing noise and ground bounce. A reasonable value for this capacitor would be about 470 nF.

The digital inputs need to be protected from excess currents (e.g. caused by induced voltage spikes) by series resistors in the range of 10 k Ω .

BAB V

PENUTUP

1.1 Kesimpulan

Dari hasil pengukuran yang didapat dari *skateboard* listrik yang telah dibuat dapat diambil kesimpulan bahwa pengendalian kecepatan menggunakan sensor *accelerometer* pada *smartphone* didapatkan respon yang cepat dan akurat. Hal itu dapat dilihat pada pengujian aplikasi pada *smartphone* yang berbeda, tergantung dari performa *smartphone* yang digunakan. *Skateboard* ini juga dapat melaju hingga kecepatannya 10,5 km/jam.

1.2 Saran

Saran dapat diberikan untuk penelitian selanjutnya dapat diberikan pengendali arah *skateboard* sehingga ketika berbelok tidak dilakukan secara manual. Selain itu agar roda dapat berputar secara optimal diperlukan pemasangan bearing pada masing-masing roda.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Ariadi. M , " Rancang Bangun *Skateboard* Listrik (Electric Board)", *Tugas Akhir*, Politeknik Negeri Bengkalis, Bengkalis, 2014.
- [2] Louise J.F."Remote control electric powered skateboard", *United States Patent*, pp. 5893425, US, April, 1999.
- [3] _____, ATmega 16, http://www.academia.edu/5225096/Atmega_16, diakses pada tanggal 21 Maret 2016.
- [4] Lister. Mesin dan Rangkaian Listrik.Jakarta:Erlangga. 1988
- [5] Achmad, Daud Umar, “Aplikasi *Accelerometer* Pada Smartphone Sebagai Input Robot Mobile”, *Tugas Akhir*, Institut Teknologi Sepuluh Nopember, Surabaya, 2015.
- [6] _____,”Modul bluetooth HC-05”.Politeknik Negeri Sriwijaya, Palembang, 2010.
- [7] Syahrul. Pemrograman Mikrokontroler AVR Bahasa Assembly dan C. Bandung: Informatika Bandung. 2014.
- [8] Solar Surya Indonesia, VRLA Baterai, <http://solarsuryaindonesia.com/info/vrla-baterai>, diakses pada tanggal 30 Maret 2016.
- [9] Anugrah Febriani, "Rancang Bangun Sistem Peringatan Keamanan Serta Pengaman Kunci Kontak Mobil Menggunakan Sensor Sidik Jari Dan Fasilitas Sms", *Tugas Akhir*, Universitas Brawijaya, Malang, 2014.
- [10] Jun Riandi,”Rancang Bangun Aplikasi Pengenalan Aktivitas Berbasis Sensor *Accelerometer* menggunakan klasifikasi k-nearest neighbours pada perangkat Mobile Berbasis Android”, *Tugas Akhir*, Institut Teknologi Sepuluh Nopember, Surabaya, 2012.
- [11] Ilyas Renreng, " Rancang Bangun Dongkrak Elektrik Kapasitas 1 Ton ", *Tugas Akhir*, Universitas Hasanuddin, Makassar, 2012.
- [12] David Wolber. App Inventor 2. United States of America: O'Reilly Media. 2011.

-----Halaman ini sengaja dikosongkan-----

DAFTAR RIWAYAT HIDUP



Nama : Stefanus Aldi Atmaja
TTL : Lumajang, 7 September 1995
Jenis Kelamin : Laki - Laki
Agama : Kristen
Alamat : Tenggilis Mejoyo AC 27
Telp/HP : 082232072226
E-mail : stevanus.aldy@yahoo.co.id

RIWAYAT PENDIDIKAN

1. 2001 – 2007 : SD Gunung Anyar 273
2. 2007 – 2010 : SMP Negeri 1 Yosowilangun
3. 2010 – 2013 : SMA Negeri 2 Lumajang
4. 2013 – 2016 : D3 Teknik Elektro, Program Studi Teknik Elektro Komputer Kontrol-FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT.PLN (Persero) Area Pengatur Distribusi Jawa Timur

PENGALAMAN ORGANISASI

1. Staff Departemen Kesejahteraan Mahasiswa Periode 2014/2015 HIMAD3TEKTRO, FTI - ITS
2. Kabiرو Akademik Departemen Kesejahteraan Mahasiswa Periode 2015/2016 HIMAD3TEKTRO, FTI - ITS

-----Halaman ini sengaja dikosongkan-----

DAFTAR RIWAYAT HIDUP



Nama : Annisa Yunitasari Firstiana
TTL : Kediri, 11 Juni 1995
Jenis Kelamin : Perempuan
Agama : Islam
Alamat : Jl. Sukarno Hatta II No.34
Kediri
Telp/HP : 085735389388
E-mail : annisafirstiana@ymail.com

RIWAYAT PENDIDIKAN

1. 2001 – 2007 : SDS Pawyatan Daha 2 Kediri
2. 2007 – 2010 : SMP Negeri 1 Kediri
3. 2010 – 2013 : SMA Negeri 7 Kediri
4. 2013 – 2016 : D3 Teknik Elektro, Program Studi Teknik Elektro Komputer Kontrol-FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT. Telekomunikasi Indonesia Divisi Regional V Jawa Timur

PENGALAMAN ORGANISASI

1. Staff Departemen Kewirausahaan Periode 2014/2015 HIMAD3TEKTRO, FTI - ITS
2. Kabirol Manajemen Bisnis Departemen Kewirausahaan Periode 2015/2016 HIMAD3TEKTRO, FTI - ITS